

10/622,572

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 7 月 1 6 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 1 9 7 8 5 0
Application Number:

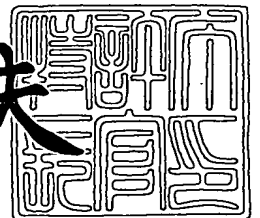
[ST. 10/C]: [J P 2 0 0 3 - 1 9 7 8 5 0]

出 願 人 株式会社リコー
Applicant(s):

2 0 0 3 年 8 月 7 日

特 許 庁 長 官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 6 3 5 1 3

【書類名】 特許願

【整理番号】 0304559

【提出日】 平成15年 7月16日

【あて先】 特許庁長官 今井 康夫 殿

【国際特許分類】 G03G 21/00

【発明の名称】 情報処理装置及び情報処理方法

【請求項の数】 15

【発明者】

 【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内

 【氏名】 高橋 征司

【特許出願人】

 【識別番号】 000006747

 【氏名又は名称】 株式会社リコー

【代理人】

 【識別番号】 100070150

 【弁理士】

 【氏名又は名称】 伊東 忠彦

【先の出願に基づく優先権主張】

 【出願番号】 特願2002-212300

 【出願日】 平成14年 7月22日

【手数料の表示】

 【予納台帳番号】 002989

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9911477

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置及び情報処理方法

【特許請求の範囲】

【請求項 1】 ネットワークを介して接続される端末から送信される W e b 情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該 W e b 情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準 W e b 情報を生成する基準 W e b 情報生成手段と、

前記第一要求に対する応答として前記基準 W e b 情報を前記端末へ送信し、前記基準パスによって特定される前記 W e b 情報を要求する第二要求を受信する通信手段とを有することを特徴とする情報処理装置。

【請求項 2】 前記通信手段は、前記ネットワークから前記第一要求を受信した場合、該第一要求に対する前記 W e b 情報へのパスにおいて、前記端末種別のデフォルト値と前記基準 W e b 情報生成手段を識別する基準 W e b 情報識別子とを付加し、

前記基準 W e b 情報生成手段は、前記通信手段によって付加された前記基準 W e b 情報識別子によって実行され、前記特定した端末種別で前記デフォルト値を置き換えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 前記通信手段は、前記第一要求に対する前記 W e b 情報へのパスにおいて、該 W e b 情報を識別する前記 W e b 情報識別子より前に前記デフォルト値を付加して前記基準パスを作成することを特徴とする請求項 1 又は 2 記載の情報処理装置。

【請求項 4】 前記 W e b 情報を生成する W e b 情報生成手段と、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記 W e b 情報を該端末に表示するための表示形式で記述することによって、 W e b ページを生成する表示情報生成手段とを有することを特徴とする請求項 1 乃至 3 のいずれか一項記載の情報処理装置。

【請求項 5】 前記表示情報生成手段は、

前記 W e b 情報生成手段によって生成された前記 W e b 情報と、前記端末種別

情報とをXMLで記述するXML記述手段と、

前記端末種別情報に基づいて、前記XMLによって記述された前記Web情報を該Web情報に応じたスタイルシートに従ってHTMLに変換することによって、前記Webページを生成するHTML変換手段とを有することを特徴とする請求項1乃至4のいずれか一項記載の情報処理装置。

【請求項6】 前記Web情報から相対パスによってリンクされる該Web情報とは異なる他のWeb情報を生成する複数の他のWeb情報生成手段を有し、

前記Web情報が表示される前記端末にてユーザによって選択されることによって要求される前記他のWeb情報の第三要求に応じて、該他のWeb情報に対応する前記他のWeb情報生成手段が該他のWeb情報を生成すると、前記表示情報生成手段は、前記基準パスに設定されている前記端末種別情報に基づいて、前記端末に応じた前記他のWeb情報を該端末に表示するWebページを生成することを特徴とする請求項1乃至5のいずれか一項記載の情報処理装置。

【請求項7】 複数のフレーム毎に表示すべき前記Web情報及び前記他のWeb情報の相対パスを設定し、Webページを分割する該複数のフレームを定義したWebフレーム情報を生成するWebフレーム情報生成手段を有し、

前記基準Web情報生成手段は、前記端末種別情報を前記Webフレーム情報へのパスに付加することにより作成した前記基準パスを含み、該端末から自動的に該パスへアクセスさせる前記基準Web情報を生成し、

前記通信手段が、前記基準Web情報を前記端末の前記第一要求に対する応答として送信し、前記基準パスによって該端末から前記Webフレーム情報を要求する前記第二要求を受信すると、前記Webフレーム情報生成手段を実行することを特徴とする請求項6記載の情報処理装置。

【請求項8】 前記表示情報生成手段は、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて前記Webフレーム情報生成手段を無効とし、前記第一要求によって要求された前記Web情報の相対パスによって前記端末から該Web情報を直接アクセスする前記Webページを生成することを特徴とする請求項7記載の情報

処理装置。

【請求項 9】 画像を形成する画像形成手段と、

前記画像形成手段を制御する画像形成制御手段とを有し、

前記 W e b 情報生成手段と前記他の W e b 情報生成手段との少なくとも 1 つの W e b 情報生成手段は、前記画像形成制御手段から前記画像形成手段に関する情報を取得して、その取得した情報に基づいて前記 W e b 情報を生成することを特徴とする請求項 6 乃至 8 のいずれか一項記載の情報処理装置。

【請求項 1 0】 前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記 W e b 情報生成手段又は前記他の W e b 情報生成手段によって生成された前記 W e b 情報又は前記他の W e b 情報にイメージを付加した前記 W e b ページを生成することを特徴とする請求項 5 乃至 8 のいずれか一項記載の情報処理装置。

【請求項 1 1】 前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記 W e b 情報生成手段又は前記他の W e b 情報生成手段によって生成された前記 W e b 情報又は前記他の W e b 情報を前記端末に対応した文字サイズで表示する前記 W e b ページを生成することを特徴とする請求項 6 乃至 1 0 のいずれか一項記載の情報処理装置。

【請求項 1 2】 前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記 W e b 情報生成手段又は前記他の W e b 情報生成手段によって生成された前記 W e b 情報又は前記他の W e b 情報を前記端末に対応した文字数で表示する前記 W e b ページを生成することを特徴とする請求項 6 乃至 1 1 のいずれか一項記載の情報処理装置。

【請求項 1 3】 前記基準 W e b 情報生成手段と、前記 W e b 情報生成手段と、前記他の W e b 情報生成手段と、前記 W e b フレーム情報生成手段とは、C 言語によって開発されたプログラムであることを特徴とする請求項 7 乃至 1 2 のいずれか一項記載の情報処理装置。

【請求項 1 4】 ネットワークを介して接続される端末から送信される W e

b 情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該 Web 情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準 Web 情報を生成する基準 Web 情報生成手順と、

前記第一要求に対する応答として送信する基準 Web 情報を前記端末へ送信し、前記基準パスによって特定される前記 Web 情報を要求する第二要求を受信する通信手順とを有することを特徴とする情報処理方法。

【請求項 15】 前記 Web 情報を生成する Web 情報生成手順と、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記 Web 情報を該端末に表示するための表示形式で記述することによって、Web ページを生成する表示情報生成手順とを有することを特徴とする請求項 14 記載の情報処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、Web アプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数の Web アプリケーションを有する情報処理装置を提供するものである。

【0002】

【従来の技術】

近年、インターネットを利用した情報提供が、クライアント PC (Personal Computer) のみならず、携帯電話器及び PDA (Personal Digital Assistants) 端末等の移動体端末へ拡大し、また、その情報提供の方法が多様化してきている。

【0003】

クライアント PC、携帯電話器及び PDA 端末等の端末側での表示画面のサイズに応じた情報提供をするために、情報提供用に予め用意してある HTML (HyperText Markup Language) データに対して、端末の種別に応じた表示内容を生成

することが提案されている（例えば、特許文献 1 参照。）。或いは、情報提供を行う W e b サーバと情報要求を行う端末とを仲介するゲートウェイサーバが、端末からの情報要求に対する W e b サーバから提供される情報をその端末の種別に応じて不要なデータ除去や画像の色属性の調整作業といった付加的作業を行うことが提案されている（例えば、特許文献 2 参照。）。

【 0 0 0 4 】

一方、近年、画像形成に関する情報を提供可能とする W e b サーバ機能を有するデータ処理装置が提案されている（例えば、特許文献 3 参照。）。このデータ処理装置によって、ユーザはインターネットを介すのみでデータ処理装置に備えられた画像形成処理を行う機器に関する情報の提供を受けることができる。

【 0 0 0 5 】

また、複数のページで構成された情報提供において、クライアント P C 側にてページ遷移が行われた場合でも、クライアント P C を利用しているユーザが入力した情報を所定領域に格納し、以後ページが遷移した際にその入力情報を用いることによって継承させて情報を提供するデータ継承方式が提案されている（例えば、特許文献 4 参照。）。

【 0 0 0 6 】

【特許文献 1】

特開平 1 1 - 1 7 5 5 1 5 号公報

【 0 0 0 7 】

【特許文献 2】

特開 2 0 0 2 - 2 3 1 0 8 号公報

【 0 0 0 8 】

【特許文献 3】

特開 2 0 0 2 - 7 0 9 5 号公報

【 0 0 0 9 】

【特許文献 4】

特開平 1 1 - 3 0 6 0 7 0 号公報

【 0 0 1 0 】

【発明が解決しようとする課題】

しかしながら、上記従来のような情報提供の方法には、以下のような問題があった。

【0011】

画像形成に関する情報を提供可能とするWebサーバ機能を有するWebサーバにおいて、画像形成処理又はその画像形成処理を行うWebサーバによって制御されるプロッタ等の機器の状態を情報として提供するような場合、その時点での状態であるため、予めHTMLデータを用意しておくことができない。したがって、特許文献1で提案されている方法を適用することができない。また、特許文献2では、ゲートウェイサーバを備えるための費用及びメンテナンス等がかかってしまう。

【0012】

特許文献3において提案されるデータ処理装置では、クライアントPC、携帯電話器及びPDA端末等の端末側での表示画面のサイズに応じた情報提供を行うことができない。

【0013】

更に、特許文献4において提案されるデータ継承方式では、ユーザに関する情報を入力させるためのページが必要であったり、ユーザが明示的にそのページへ入力する必要がある。また、ネットワーク接続されている間ユーザ毎に継承させるべき情報を管理しなければならない。従って、表示画面のサイズに応じた情報提供を自動的に行うことができない。

【0014】

そこで、本発明の課題は、Webアプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数のWebアプリケーションを有する情報処理装置を提供することである。

【0015】**【課題を解決するための手段】**

上記課題を解決するため、本発明は、請求項1に記載されるように、ネットワ

ークを介して接続される端末から送信されるW e b情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該W e b情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準W e b情報を生成する基準W e b情報生成手段と、前記第一要求に対する応答として送信する基準W e b情報を前記端末へ送信し、前記基準パスによって特定される前記W e b情報を要求する第二要求を受信する通信手段とを有するように構成される。

【 0 0 1 6 】

このような情報処理装置では、端末から基準パスで自動的にアクセスさせるW e b情報を生成して提供するため、端末側では、ユーザによる端末種別情報の入力を行なうことなく、このW e b情報によって端末種別情報を含む基準パスを取得することができる。また、ユーザが求めるW e b情報を提供する前に基準パスが提供されるため、W e b情報を生成するW e bアプリケーションの開発者は、端末種別毎の開発を行わなくてよい。

【 0 0 1 7 】

前記W e b情報は、例えば、インターネットを介してW e bブラウザ上で提供される情報である。

【 0 0 1 8 】

更に、本発明は、請求項 2 に記載されるように、前記通信手段は、前記ネットワークから前記第一要求を受信した場合、該第一要求に対する該W e b情報へのパスにおいて、前記端末種別のデフォルト値と前記基準W e b情報生成手段を識別する基準W e b情報識別子とを付加し、前記基準W e b情報生成手段は、前記通信手段によって付加された前記基準W e b情報識別子によって前記基準W e b情報生成手段が実行され、該特定した端末種別で前記デフォルト値を置き換えるように構成することができる。

【 0 0 1 9 】

このような情報処理装置では、ネットワークから第一要求を受信すると、通信手段（例えば、H T T Pデーモン）が予めデフォルト値と基準W e b情報識別子とを付加するため、第一要求に対して常に基準W e b情報生成手段を実行させる

ことができる。

【0020】

また、本発明は、請求項3に記載されるように、前記通信手段は、前記第一要求に対する前記Web情報へのパスにおいて、該Web情報を識別する前記Web情報識別子より前に前記デフォルト値を付加して前記基準パスを作成するように構成することができる。

【0021】

このような情報処理装置では、端末種別情報がWeb情報識別子より前に設定されるため、Web情報識別子以降を相対パスとして設定されるようにすることができる。従って、基準Web情報からページ遷移した場合において、常に端末種別情報が継承される。

【0022】

更に、本発明は、請求項4に記載されるように、前記Web情報を生成するWeb情報生成手段と、前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記Web情報を該端末に表示するための表示形式で記述することによって、Webページを生成する表示情報生成手段とを有するように構成することができる。また、本発明は、請求項5に記載されるように、前記表示情報生成手段は、前記Web情報生成手段によって生成された前記Web情報と、前記端末種別情報とをXMLで記述するXML記述手段と、前記端末種別情報に基づいて、前記XMLによって記述された前記Web情報を該Web情報に応じたスタイルシートに従ってHTMLに変換することによって、前記Webページを生成するHTML変換手段とを有するように構成できる。

【0023】

このような情報処理装置では、表示情報生成手段によって端末種別に対応したWebページの生成が行われ、また、Web情報の生成に依存しないため、Webアプリケーションの追加を容易に実現することができる。

【0024】

また、本発明は、請求項6に記載されるように、前記Web情報から相対パス

によってリンクされる該Web情報とは異なる他のWeb情報を生成する複数の他のWeb情報生成手段を有し、前記Web情報が表示される前記端末にてユーザによって選択されることによって要求される前記他のWeb情報の第三要求に応じて、該他のWeb情報に対応する前記他のWeb情報生成手段が該他のWeb情報を生成すると、前記表示情報生成手段は、前記基準パスに設定される前記端末種別情報に基づいて、前記端末に応じた前記他のWeb情報を該端末に表示するWebページを生成するように構成することができる。

【0025】

このような情報処理装置において、端末へ提供されたWeb情報から他のWeb情報が相対パスによってリンクされた場合においても、基準パスに基づいて共通に設定される端末種別情報を常に取得することができるため、常に、端末の表示画面に応じたWebページ生成することができる。

【0026】

更に、本発明は、請求項7に記載されるように、複数のフレーム毎に表示すべき前記Web情報及び前記他のWeb情報の相対パスを設定し、Webページを分割する該複数のフレームを定義したWebフレーム情報を生成するWebフレーム情報生成手段を有し、前記基準Web情報生成手段は、前記端末種別情報を前記Webフレーム情報へのパスに付加することにより作成した前記基準パスを含み、該端末から自動的に該パスへアクセスさせる前記基準Web情報を生成し、前記通信手段が、前記基準Web情報を前記端末の前記第一要求に対する応答として送信し、前記基準パスによって該端末から前記Webフレーム情報を要求する前記第二要求を受信すると、前記Webフレーム情報生成手段を実行するように構成することができる。

【0027】

このような情報処理装置において、端末にフレームによって分割されたWebページを提供する場合においても、基準パスを設定したWebフレーム情報を端末側から要求するようにするため、各フレームからアクセスされるWeb情報及び他のWeb情報は相対パスによってアクセスすることができ、且つ、端末種別情報を継承させることができる。

【0028】

また、本発明は、請求項8に記載されるように、前記表示情報生成手段は、前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記Webフレーム情報生成手段を無効とし、前記第一要求によって要求された前記Web情報の相対パスによって前記端末側から該Web情報が直接アクセスする前記Webページを生成するように構成することができる。

【0029】

このような情報処理装置において、端末がPC以外の場合にはフレーム情報を端末に送信せず直接Web情報をアクセスするWebページ生成して送信することができる。

【0030】

更に、本発明は、請求項9に記載されるように、画像を形成する画像形成手段と、前記画像形成手段を制御する画像形成制御手段とを有し、前記Web情報生成手段と前記他のWeb情報生成手段との少なくとも1つのWeb情報生成手段は、前記画像形成制御手段から前記画像形成手段に関する情報を取得して、その取得した情報の基づいて前記Web情報を生成するように構成することができる。

【0031】

このような情報処理装置において、端末種別情報を継承しつつ、情報処理装置に実装される画像形成するプロッタ又はスキャナ等の機器のステータスをWeb情報として提供することができる。

【0032】

また、本発明は、請求項10に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Web情報生成手段又は前記他のWeb情報生成手段によって生成された前記Web情報又は前記他のWeb情報にイメージを付加した前記Webページを生成するように構成することができる。

【0033】

このような情報処理装置において、端末の表示画面が例えばPCより小さい場

合、イメージの表示を行わないようにWebページを生成することができる。また、端末の表示画面がPCより大きい場合、より大きいイメージ又は複数のイメージを表示するようにWebページを生成することができる。

【0034】

更に、本発明は、請求項11に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Web情報生成手段又は前記他のWeb情報生成手段によって生成された前記Web情報又は前記他のWeb情報を前記端末に対応した文字サイズで表示する前記Webページを生成するように構成することができる。

【0035】

このような情報処理装置において、端末の表示画面が例えばPCより小さい場合、小さい文字サイズで表示するWebページを生成することができる。また、端末の表示画面がPCより大きい場合、大きい文字サイズで表示するWebページを生成することができる。

【0036】

また、本発明は、請求項12に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて、前記Web情報生成手段又は前記他のWeb情報生成手段によって生成された前記Web情報又は前記他のWeb情報を前記端末に対応した文字数で表示する前記Webページを生成するように構成することができる。

【0037】

このような情報処理装置において、端末の表示画面が例えばPCより小さい場合、少ない文字数で表示するWebページを生成することができる。また、端末の表示画面がPCより大きい場合、文字数を多くして詳細にしたWebページを生成することができる。

【0038】

更に、前記課題を解決するための手段として、本発明は、上記情報処理装置における処理をコンピュータに行なわせるための情報処理方法とすることもできる。

【0039】**【発明の実施の形態】**

以下、本発明の実施の形態を図面に基づいて説明する。

【0040】

本発明の一実施例に係る情報処理装置は、プリンタ、FAX、コピー等の複数の異なる画像形成機能の少なくとも1つを有すると共に、複数のWebアプリケーションによって画像形成に関する情報を提供する。本実施例において、説明の便宜上、主に、印刷処理又は印刷を行うプロッタに関する情報が提供される場合について説明する。

【0041】

図1は、本発明の一実施例に係る情報処理装置のハードウェア構成を示すブロック図である。図1において、情報処理装置100は、コンピュータによって制御される装置であって、CPU（中央処理装置）11と、ROM（Read-Only Memory）12と、RAM（Random Access Memory）13と、不揮発性RAM（non-volatile Random Access Memory）14と、リアルタイムクロック15、イーサネット（登録商標）I/F（Ethernet（登録商標）Interface）21と、USB（Universal Serial Bus）22と、IEEE（Institute of Electrical and Electronics Engineers）1284 23と、ハードディスクI/F 24と、エンジンI/F 25と、RS-232C I/F 26とで構成され、システムバスBに接続される。

【0042】

CPU 11は、ROM 12に格納されたプログラムに従って情報処理装置100を制御する。RAM 13には、例えば、各インターフェース21から26に接続される資源に領域が割り当てられる。不揮発性RAM 14には、情報処理装置100を制御するためにCPU 11による処理に必要な情報が格納される。リアルタイムクロック15は、現時刻を計ると共に、処理を同期させる場合にCPU 11によって使用される。

【0043】

イーサネット（登録商標）I/F 21には、10BASE-T又は100BA

S E-T X等のイーサネット（登録商標）用インターフェースケーブルが接続される。U S B 2 2 には、U S B 用インターフェースケーブルが接続される。I E E E 1284 2 3 には、I E E E 1284 用インターフェースケーブルが接続される。

【0044】

ハードディスク I / F 2 4 には、ハードディスク 3 4 が接続され、ネットワークを介して送信された印刷すべき文書の文書データ、又は、印刷処理後の画像データがハードディスク I / F 2 4 を介してハードディスク 3 4 に格納される。エンジン I / F 2 5 には、文書データに基づいて所定媒体に印刷を行うプロッタ 3 5-1 及び画像データを取り込むスキャナ 3 5-2 等が接続される。R S-2 3 2 C I / F 2 6 には、オペレーションパネル 3 6 が接続され、ユーザへの情報の表示及びユーザから入力情報又は設定情報の取得が行われる。

【0045】

次に、図 1 に示すようなハードウェア構成を有し、複数の異なる画像形成処理を可能とし、かつ、複数の W e b アプリケーションを有する情報処理装置 1 0 0 の機能構成について説明する。

【0046】

図 2 は、情報処理装置の機能構成を示すブロック図である。図 2 において、情報処理装置 1 0 0 は、インターネット 1 6 を介して表示画面の利用形態の異なるクライアント P C 4 1、携帯電話器 4 2 及び P D A (Personal Digital Assistants) 端末 4 3 等の端末 4 0 と接続可能であって、各端末 4 0 からの要求に応じて、その要求に対する応答として情報を提供するコンピュータである。

【0047】

情報処理装置 1 0 0 は、主に、ネットワーク制御部 1 0 1 と、シーケンス制御ライブラリ 1 1 0 と、W e b アプリ振分共通ライブラリ 1 2 0 と、W e b ページハンドラ 2 0 0 と、S O A P (Simple Object Access Protocol) ライブラリ 2 0 1 と、X M L (eXtensible Markup Language) ライブラリ 2 0 3 と、X S L T (XSL Transformations) プロセッサ 2 0 5 と、W e b ページ機能 (W P F) 3 0 0 と、プリンタ制御部 1 0 3 と、スキャナ制御部 1 0 5 とを有する。

【0048】

ネットワーク制御部 1 0 1 は、少なくとも H T T P (Hyper Text Transfer Protocol) による通信制御を行う H T T P デーモン 2 を有し、H T T P に従って端末 4 0 から要求を受信し、その要求に対する応答としてその要求に応じた情報提供を行う。

【 0 0 4 9 】

H T T P デーモン 2 は、端末 4 0 から要求を受信すると、複数の W e b アプリケーション 3 0 1 において、ページ遷移した際にも端末 4 0 の表示画面での利用形態に関する属性を示す属性情報が継承されるようにする所定の処理を実行するために、要求で指定される U R L (Uniform Resource Locator) に仮の共通パス情報と、上記所定の処理を実行するための W e b アプリケーション I D と C G I (Common Gateway Interface) とを付加する。ページ遷移とは、表示されている W e b ページから、その W e b ページにリンクされる他の W e b ページを表示させることを言う。

【 0 0 5 0 】

H T T P デーモン 2 は、必要な情報を付加して、図 3 に示すような U R L を作成する。図 3 は、U R L 構成を示す図である。図 3 において、U R L は、大きく共通パス情報 6 0 a と相対パス情報 6 0 b とで構成される。H T T P デーモン 2 が端末 4 0 から最初のページ要求を受信した際には、例えば、「http://AAA」のみが U R L として指定されており、この U R L に対して、H T T P デーモン 2 は、所定の「/TERMINAL/PROFILE/LANG/webDefaultApl/default.cgi」を付加する。

【 0 0 5 1 】

このようにして構成された U R L は、例えば、「http」等のアクセス手段を特定するプロトコル種別 6 1 と、「AAA」で示される I P アドレス 6 2 と、「TERMINAL」で示される端末種別情報 6 3 a と、「PROFILE」で示されるプロファイル情報 6 3 b と、「LANG」で示される言語情報 6 3 c と、「webDefaultApl」で示される W e b アプリケーション I D 6 4 と、「default.cgi」で示されるページ情報 6 5 とを有する。

【 0 0 5 2 】

そして、プロトコル種別 6 1 と、I P アドレス 6 2 と、端末種別情報 6 3 a と

、プロファイル情報 6 3 b と、言語情報 6 3 c とが共通パス情報 6 0 a として、以後のページ遷移において継承され、ページ遷移によって跨る W e b アプリケーション 3 0 1 との間で参照される。また、W e b アプリケーション I D 6 4 と、ページ情報 6 5 とが相対パス情報 6 0 b を構成する。

【 0 0 5 3 】

図 2 に戻り、処理部の説明を続ける。

【 0 0 5 4 】

シーケンス制御ライブラリ 1 1 0 は、インターネット 1 6 を介して行われるデータの送受信の処理シーケンスと各 W e b アプリケーション 3 0 1 とのデータの受け渡しの処理シーケンスとの違いを所定のシーケンス制御処理によって吸収する、複数の W e b アプリケーション 3 0 1 に対して共通の処理部である。

【 0 0 5 5 】

W e b アプリ振分共通ライブラリ 1 2 0 は、端末 4 0 からの要求の解析及び端末 4 0 への応答を生成し、複数の W e b アプリケーション 3 0 1 に対して共通の処理部である。W e b アプリ振分共通ライブラリ 1 2 0 は、W e b ページハンドラ 2 0 0 にて XML で記述された応答を、X S L T プロセッサ 2 0 5 によって各端末 4 0 の端末種別に応じた H T M L (HyperText Markup Language) による表示形式に変換する。

【 0 0 5 6 】

W e b ページハンドラ 2 0 0 は、W e b アプリケーション 3 0 1 が解釈可能な処理言語と、端末 4 0 との間で行われる要求及び応答による通信制御で解釈される処理言語との変換を行う処理部である。W e b ページハンドラ 2 0 0 は、要求に対応する W e b アプリケーション 3 0 1 を C G I を介して関数コールする。また、W e b ページハンドラ 2 0 0 は、W e b ページ機能 3 0 0 から通知された処理結果データを XML で記述するために処理結果データのシリアルイズ要求を S O A P ライブラリ 2 0 1 に対して行う。

【 0 0 5 7 】

S O A P ライブラリ 2 0 1 は、C 言語の変数で与えられた処理結果データを X M L によって記述することによってデータ変換をしてシリアルイズする。本実施

例において、シリアル化するとは、XMLによってWebページ機能300から通知された処理結果を記述することである。

【0058】

XMLライブラリ203は、SOAPライブラリ201に利用されることによってXMLで処理結果をシリアル化する。また、XMLライブラリ203は、XSLTプロセッサ205に利用されることによって、処理結果を示すHTMLを生成する。

【0059】

XSLTプロセッサ205は、端末40の要求で指定されるWebアプリケーション301に対応するXSL(eXtensible Stylesheet Language)に基づいて、XMLライブラリ203を利用することによってXMLで記述された処理結果をHTMLの記述に変換する。

【0060】

Webページ機能300は、Webページハンドラ200から関数コールされると、例えば、情報処理装置100に備えられた画像形成を制御するプリンタ制御部103又はスキャナ制御部105にAPI(Application Program Interface)17を介して問い合わせ、プロッタ35-1の状態又はスキャナ35-2の状態等を示す情報を取得し、その情報をWebページハンドラ200へ返す。また、Webページ機能300は、同様に、API17を介してネットワーク制御部101へ問い合わせることによって取得した情報をWebページハンドラ200へ返す。

【0061】

次に、Webアプリケーションをするための詳細な構成を図4及び図5で説明する。図4は、Webアプリケーションの実行を説明するためのブロック図である。図5は、ハッシュリストを示す図である。図4において、Webページハンドラ200は、例えば、図3に示すようなURLの構成に基づいた情報提供が行えるように、端末40からの最初の要求時に実行されるデフォルトハンドラ210と、端末40からの指示によるページ遷移によって実行されるネットワーク設定ハンドラ220と、同様にページ遷移によって実行されるシステム構成情報提

供ハンドラ 230 等を有する。

【0062】

Web ページ機能 300 は、Web ページハンドラ 200 の各ハンドラに対応して、Web デフォルトアプリ 310 と、Web ネットワーク設定アプリ 320 と、Web システム構成情報提供アプリ 330 等の複数の Web アプリケーション 301 を有する。

【0063】

端末 40 に初期画面が表示されるまでに、デフォルトハンドラ 210 は、図 3 に示される URL の形式に従って指定されるページ情報 65 に基づいて、順に、「default.cgi」、「mainFrame.cgi」、「menuPage.cgi」、「topPage.cgi」、「headerPage.cgi」を介して Web デフォルトアプリ 310 を関数コールする。その結果、例えば、情報処理装置 100 に実装されたプロッタ 35-1 又はスキャナ 35-2 等の機器の状態が端末 40 に表示される。

【0064】

ネットワーク設定ハンドラ 220 は、端末 40 のユーザによって入力されたネットワーク設定情報を情報処理装置 100 内に設定するための Web ネットワーク設定アプリ 320 を関数コールする。また、システム構成情報提供ハンドラ 230 は、Web システム構成情報提供アプリ 330 を関数コールすることによって、端末 40 からの要求に応じて情報処理装置 100 によって成されるシステムの構成情報を取得する。

【0065】

Web アプリ振分共通ライブラリ 120 は、URL から Web アプリケーション ID 64 を取得し、図 5 (A) に示されるハッシュリスト 50a を参照して、その Web アプリケーション ID 64 が「webDefaultApl」を示す場合、「default_handler」を実行する。Web アプリケーション ID 64 が他の Web アプリケーション ID を示す場合についても同様に、ハッシュリスト 50a を参照して、実行すべき Web アプリケーションに対応するハンドラを実行する。

【0066】

また、Web ページハンドラ 200 の各ハンドラ 210、220 及び 230 は

、指定されたURLからページ情報65を取得し、図5（B）に示されるハッシュリスト50bを参照して、対応する関数をコールする。例えば、ページ情報65が「topPage.cgi」を示す場合、ハッシュリスト50bから「topPage()」をコールすべき関数として取得する。ページ情報65が他のCGIを示す場合についても同様に、ハッシュリスト50bを参照して、コールすべき関数を取得する。

【0067】

更に、Webアプリ振分共通ライブラリ120は、XSLTプロセッサ205に対してXMLで記述された処理結果をHTMLに変換するXSL変換要求を行うために、Webページ機能IDとして指定されたURLからページ情報65を取得し、図5（B）に示されるハッシュリスト50bを参照して、対応するスタイルシートを取得する。例えば、ページ情報65が「topPage.cgi」を示す場合、Webアプリ振分共通ライブラリ120は、ハッシュリスト50bから「topPage.xsl」をXSL変換に必要なスタイルシートとして取得する。ページ情報65が他のCGIを示す場合についても同様に、ハッシュリスト50bを参照して、XSL変換に必要なスタイルシートを取得する。

【0068】

図4及び図5（B）において、Webページ機能IDとしてのURLのページ情報65が「default.cgi」である場合、default()関数が実行され、端末40の端末種別が決定され、HTTPデーモン2によって仮に作成されたURLの端末種別情報にその決定した端末種別が設定された初期画面が作成される。

【0069】

ページ情報65が「mainFrame.cgi」である場合、mainFrame()関数が実行され、初期画面を構成するフレーム毎に、呼び出すべき所定のCGIが設定される。例えば、mainFrame()関数によって、メニューページを表示するフレームと、トップページを表示するフレームと、ヘッダページを表示するフレームとが構成されるような情報が生成される。

【0070】

ページ情報65が「menuPage.cgi」である場合、menuPage()関数が実行され、メニューページを作成するための情報が生成される。ページ情報65が「topPag

e.cgi」である場合、topPage()関数が実行され、例えば、プロッタ 3 5 - 1 の状態がプリンタの状態として表示するトップページが作成するための情報が生成される。ページ情報 6 5 が「headerPage.cgi」である場合、headerPage()関数が実行され、ヘッダページを作成するための情報が生成される。

【 0 0 7 1 】

また、ページ情報 6 5 が「netsetting.cgi」である場合、setting()関数が実行され、システム設定を行うためのページを作成するための情報が生成される。ページ情報 6 5 が「sysconfig.cgi」である場合、config()関数が実行され、システム構成を示す情報が生成される。

【 0 0 7 2 】

説明の便宜上、WebアプリケーションIDの名称とWebページハンドラの名称とを統一させているが、このような名称に限定されるものではなく、図 5 (A) のようなハッシュリスト 5 0 a によって対応付けがなされれば良い。同様に、Webページ機能IDと、関数と、スタイルシートの名称を統一させているが、このような名称に限定されるものではなく、図 5 (B) のようなハッシュリスト 5 0 b によって対応付けがなされれば良い。

【 0 0 7 3 】

次に、端末 4 0 からの最初の要求に対して情報提供が行われるまでの処理フローについて説明する。図 6 及び図 7 は、default.cgi による処理フローを示す図である。

【 0 0 7 4 】

図 6 において、端末 4 0 は、HTTPのGETメソッドを使用して、ユーザエージェント情報を付加した要求を行う（ステップ S 1 1）。端末 4 0 は、例えば、URL「http://AAA」を指定することによってその要求を行う。

【 0 0 7 5 】

この要求を受信した情報処理装置 1 0 0 のHTTPデーモン 2 は、URL「http://AAA」に端末種別情報 6 3 a と、プロファイル情報 6 3 b と、言語情報 6 3 c と、WebアプリケーションID 6 4 と、ページ情報 6 5 とを付加してシーケンス制御ライブラリ 1 1 0 へこの要求を通知する（ステップ S 1 2）。例えば、

HTTPデーモン2によって、端末種別情報63aとして「TERMINAL」、プロファイル情報63bとして「PROFILE」、言語情報63cとして「LANG」、WebアプリケーションID64として「webDefaultApl」、ページ情報65として「default.cgi」がURL「http://AAA」に付加され、「http://AAA/TERMINAL/PROFILE/LANG/webDefaultApl/default.cgi」がURLとして構成される。

【0076】

シーケンス制御ライブラリ110は、Webアプリ振分共通ライブラリ120を呼び出す（ステップS13）。その際、シーケンス制御ライブラリ110は、端末40との接続に関する情報を示すHTTP接続情報と、ハッシュリスト50a及び50b（以下、ハッシュリスト50a及び50bを総称してハッシュリストと言う）とを引数として設定する。このHTTP接続情報は、接続が切断されるまで端末40との接続を一意に特定する。

【0077】

Webアプリ振分共通ライブラリ120は、HTTPデーモン2によって生成されたURLに基づいて、「webDefaultApl」を示すWebアプリケーションID64に対応するWebページハンドラ200として、ハッシュリスト50aからデフォルトハンドラ210（「default_handler」）を特定して呼び出す（ステップS14）。

【0078】

Webページハンドラ200では、デフォルトハンドラ210が、URLから「default.cgi」を示すページ情報65を取得し、その「default.cgi」に対応するWebページ機能300として、ハッシュリスト50bからdefault()関数を特定して呼び出す（ステップS15）。この際、HTTP接続情報が引数として設定される。例えば、Webページ機能300がC言語で開発されている場合、Webページハンドラ200は、C言語の処理型に変換してdefault()関数をコールする。

【0079】

default()関数の実行によって、HTTP接続情報に基づいて、端末情報が解

析され、決定した共通パス情報 6 0 a によって初期画面を呼び出す URL が作成される（ステップ S 1 6）。その URL を含む処理結果データとして Web ページハンドラ 2 0 0 におけるデフォルトハンドラ 2 1 0 へ通知される（ステップ S 1 7）。default.cgi による処理では、共通パス情報 6 0 a を決定し、決定した共通パス情報 6 0 a と初期画面のフレーム構成を決定するための相対パス情報 6 0 b とによって URL が設定される。例えば、共通パス情報 6 0 a として「/pc/admin/ja」が決定され、相対パス情報 6 0 b として、初期画面をフレームによって幾つかの表示領域に分割する「/webDefaultApl/mainFrame.cgi」を設定する。

【 0 0 8 0 】

デフォルトハンドラ 2 1 0 は、処理結果データを XML で記述するために、シリアル化要求を SOAP ライブラリ 2 0 1 に対して行う（ステップ S 1 8）。SOAP ライブラリ 2 0 1 は、例えば、C 言語の構造体で示される処理結果データに基づいて、DOM (Document Object Model) を作成し、必要な要素及び属性を追加して（ステップ S 1 9）、XML ライブラリ 2 0 3 によって処理結果データを XML で記述する（ステップ S 2 0）。XML で記述された処理結果データは、SOAP ライブラリ 2 0 3 によってシリアル化結果としてデフォルトハンドラ 2 1 0 に通知される（ステップ S 2 1）。Web ページハンドラ 2 0 0 において、デフォルトハンドラ 2 1 0 は、通知されたシリアル化結果を端末 4 0 に表示すべき表示データ「Response DOM」として、Web アプリ振分共通ライブラリ 1 2 0 へ通知する（ステップ S 2 2）。

【 0 0 8 1 】

Web アプリ振分共通ライブラリ 1 2 0 は、XML で記述される表示データを端末 4 0 の種別に応じた HTML の記述に変換するために、ハッシュリスト 5 0 b を参照して、「default.cgi」を示すページ情報 5 6 に対応するスタイルシート「default.xsl」を取得して、XSLT プロセッサ 2 0 5 に対して XSL 変換要求を行う（ステップ S 2 3）。この際、表示データ「Response DOM」と、スタイルシート「default.xsl」とが引数として設定される。

【 0 0 8 2 】

XSL 変換要求を受信した XSLT プロセッサ 2 0 5 は、引数で指定されたス

タイルシート「default.xml」を実行することによって、XMLで記述された表示データ「Response DOM」の構文を解析し（XMLパース）、DOMを作成し、要素及び属性を追加して（ステップS24）、XMLライブラリ203によってHTMLの記述に変換する（ステップS25）。XSLTプロセッサ205は、変換したHTMLの記述をXSL変換結果としてWebアプリ振分共通ライブラリ120へ通知する（ステップS26）。

【0083】

このように、ページ情報65が「default.cgi」の場合のWebページ機能300によってHTMLで記述された処理結果が、Webアプリ振分共通ライブラリ120にデフォルトHTMLとして出力される。デフォルトHTMLでは、HTMLで「mainFrame.cgi」が呼び出されるように記述され、デフォルトHTML応答として出力される。

【0084】

このデフォルトHTML応答は、順に、Webアプリ振分共通ライブラリ120からシーケンス制御ライブラリ110へ通知され（ステップS27）、シーケンス制御ライブラリ110からHTTPデーモン2へ通知される（ステップS28）。そして、HTTPデーモン2によって、HTTPに従ってデフォルトHTML応答がステップS11における端末40の要求に対する応答として送信され（ステップS29）、default.cgiによってWebデフォルトアプリ310を実行する処理フローを終了する。

【0085】

ステップS17では、共通パス情報60aを決定し、決定した共通パス情報60aを相対パス情報60bに付加するのみであるので、Webデフォルトアプリ310にて処理結果をHTMLで記述し、そのHTMLで記述された処理結果を出力するようにし、ステップS18からS26を省略しても良い。

【0086】

次に、図6のステップS16にてdefault()関数のコールによって実行される端末情報の解析処理について図8から図11で詳述する。図8は、端末情報の解析処理について説明するためのフローチャート図である。図9は、HTTP接続

情報のデータ構造を示す図である。図10は、HTTP要求情報のデータ構造を示す図である。図11は、ヘッダ情報リストを示す図である。図9から図11に示されるデータ構造は、図8に示される端末情報の解析処理にて参照されるデータ構造である。

【0087】

図8での解析処理を説明する前に、図9から図11に示される各データ構造について説明する。図9において、HTTP接続情報340は、接続毎に生成される情報であって、「HTTP_REQUEST_INFO *request」にて示される記述346は端末40からの要求の内容へのポインタを示し、「HTTP_RESPONSE_INFO *response」にて示される記述347は端末40への応答の内容へのポインタを示す。この記述346によって指定されるポインタの先には、HTTP要求情報が図10に示すように格納されている。

【0088】

図10において、HTTP要求情報350は、リクエストをした側の端末40のアドレスを示すリモートアドレス351と、そのリクエストを受信した側のホストアドレスを示すローカルアドレス352と、POST、GET等のリクエストメソッド353と、リクエストのURLを示すリクエストURL354と、リクエストのHTTPバージョン355と、ヘッダ情報のリスト360と、エンコードされた情報のリスト357とを有する。

【0089】

リクエストのURL354には、図3に示されるような共通パス情報60aと相対パス情報60bとで構成されるURLがシーケンス制御ライブラリ110によって設定される。HTTP要求情報350は、HTTP接続情報340がWebページハンドラ200とWebページ機能300とへ通知されることによって、同時に通知される。従って、各Webページハンドラ200と各Webアプリケーション301は、共通パス情報60aを参照できる。

【0090】

そして、ヘッダ情報リスト360は、例えば、図11に示すような情報によって構成される。図11において、ヘッダ情報リスト360は、「User-Agent:Moz

illa/4.0 {compatible;MSIE 6.0;Windows (登録商標) NT 5.0;Q312461;NET CLR 1.0.3705} にて示されるユーザエージェント情報 3 6 5 を有する。このユーザエージェント情報 3 6 5 は、端末 4 0 の種別を示し、この記述を解析することによって、端末 4 0 での表示画面に応じた情報提供をすることができる。

【 0 0 9 1 】

図 8 において、default()関数のコールによって、Web デフォルトアプリ 3 1 0 は、HTTP 接続情報 3 4 0 から HTTP 要求情報 3 5 0 を取得する（ステップ S 1）。更に、Web デフォルトアプリ 3 1 0 は、HTTP 要求情報 3 5 0 からユーザエージェント情報 3 6 5 を取得する（ステップ S 2）。そして、Web デフォルトアプリ 3 1 0 は、ユーザエージェント情報 3 6 5 の Web ブラウザ種別に基づいて、Web ブラウザ種別—端末種別対応表（図 1 2）を参照して、端末 4 0 の端末種別を特定する（ステップ S 3）。

【 0 0 9 2 】

Web デフォルトアプリ 3 1 0 は、Web ブラウザ種別で端末種別が判断できたか否かを判断する（ステップ S 4）。判断できる場合、ステップ S 3 8 へ進む。判断できない場合、ユーザエージェント情報 3 6 5 の Web ブラウザ OS に基づいて、Web ブラウザ OS—端末種別対応表を参照して、端末 4 0 の端末種別を特定する（ステップ S 5）。

【 0 0 9 3 】

Web デフォルトアプリ 3 1 0 は、Web ブラウザ OS で端末種別が判断できたか否かを判断する（ステップ S 6）。判断できない場合、サポート対象外であることを示すエラー情報を出力し（ステップ S 7）、端末情報の解析処理を終了する。ステップ S 7 において、Web デフォルトアプリ 3 1 0 は、サポート対象外であることを示すエラー情報を出力する代わりに、所定の端末種別を端末 4 0 の端末種別として設定し、ステップ S 8 へ進むようにしても良い。

【 0 0 9 4 】

一方、ステップ S 4 又は S 6 にて、端末種別が判断できる場合、Web デフォルトアプリ 3 1 0 は、URL の端末種別情報 6 3 a に端末種別を設定し（ステップ S 8）、端末情報の解析処理を終了する。

【0095】

図12は、Webブラウザ種別—端末種別対応表を示す図である。図12において、Webブラウザ種別—端末種別対応表370は、Webブラウザ種別と端末種別等の項目を有し、Webブラウザ種別に応じた端末種別が管理される。

【0096】

Webブラウザ種別として、例えば、「DOCOMO（登録商標）」、「xiino（登録商標）」、「MSPIE（登録商標）」、「HandHTTP（登録商標）」、「MSIE（登録商標）」、「Mozilla（登録商標）」、「mozilla/5（登録商標）」、「Netscape6（登録商標）」等が管理される。

【0097】

端末種別として、「IMODE（登録商標）」、「PDA」、「0」、「PC」等がある。Webブラウザ種別「DOCOMO（登録商標）」に対応する端末種別は「IMODE（登録商標）」であり、Webブラウザ種別「xiino（登録商標）」、「MSPIE（登録商標）」、「HandHTTP（登録商標）」に対応する端末種別は「PDA」であり、Webブラウザ種別「Mozilla（登録商標）」、「Mozilla（登録商標）」、「mozilla/5（登録商標）」、「Netscape6（登録商標）」に対応する端末種別は「PC」である。端末種別は、上記以外に限定されるものではなく、更に多くの種別があっても良い。

【0098】

端末種別「0」は、端末種別を特定できないことを示す。図8のステップS4での判断処理において、判断できない場合に相当する。例えば、Internet Explorer（登録商標）からの要求である場合、通常、端末種別が「PDA」の場合にはユーザエージェント情報365にWebブラウザ種別「MSPIE」が設定され、端末種別が「PC」の場合にはユーザエージェント情報365にWebブラウザ種別「MSIE」が設定される。しかしながら、端末種別が「PDA」の場合であっても、ユーザエージェント情報365にWebブラウザ種別「MSIE」が設定されることがある。従って、Webブラウザ種別「MSI

E」に対応する端末種別を予め特定することができないため、「0」が設定されている。

【0099】

このように端末種別を特定できない場合、図13に示すような対応表が参照される。図13は、WebブラウザOS－端末種別対応表を示す図である。図13において、WebブラウザOS－端末種別対応表380は、WebブラウザOSと端末種別等の項目を有し、WebブラウザOSに応じた端末種別が管理される。

【0100】

WebブラウザOSとして、例えば、「Windows（登録商標）3.1」、「Windows（登録商標）95」、「Windows（登録商標）98」、「Windows（登録商標）ME」、「Windows（登録商標）NT」、「Windows（登録商標）2000」、「Windows（登録商標）XP」、「Mac 68K（登録商標）」、「Mac PowerPC（登録商標）」、「Solaris（登録商標）」、「Linux（登録商標）」、「FreeBSD（登録商標）」、「AIX（登録商標）」、「IRIX（登録商標）」、「HP（登録商標）」、「OS/2（登録商標）」、「Windows（登録商標）CE」等が管理される。

【0101】

端末種別として、「PDA」、「PC」等がある。携帯電話器42は、端末種別「PDA」に含む。図13に示すWebブラウザOS－端末種別対応表380において、WebブラウザOS「Windows（登録商標）3.1」から「OS/2（登録商標）」に対応する端末種別は「PC」であり、WebブラウザOS「Windows（登録商標）CE」に対応する端末種別は「PDA」である。

【0102】

このようにWebブラウザ種別－端末種別対応表370とWebブラウザOS－端末種別対応表380とによって端末種別を特定することができる。

【0103】

default.cgiによるデフォルトHTML出力について図14で説明する。図14は、デフォルトHTML出力の例を示す図である。図14において

、デフォルトHTML 4 0 0 は、端末 4 0 にて表示されることはなく、J a v a（登録商標）S c r i p t（登録商標）の記述 4 0 1 によって、トップページを呼び出すためのスクリプトが実行される。

【0 1 0 4】

「self.document.cookie=” cookieOnOffchecker=on; path=/"」で示される記述 4 0 2 によって、端末 4 0 のWebブラウザの情報をc o o k i eによって取得し、「self.location.pathname=” /pc/admin/ja/webDefaultApl/mainframe.cgi”」で示される記述 4 0 3 によって、端末種別「PC」を継承して、Webページ機能 3 0 0 のWebデフォルトアプリ 3 1 0 が実行される、つまり、「mainframe.cgi」によってメインフレームHTMLを要求する。記述 4 0 3 によって、以後ユーザによって行われる一連のページ遷移において、<a href>タグで指定される相対パスの基準となる絶対パスが指定されたことになる。

【0 1 0 5】

尚、インターネット 1 6 を介してクライアントPC 4 1 に情報が提供される場合、画面をフレームによって複数に分割することがよく行われる。図 1 4 では、そのような場合のデフォルトHTML出力の例を示している。一方、フレームによって画面を分割しないで初期画面を表示する場合は、例えば、記述 4 0 3 は、「self.location.pathname=” /pc/admin/ja/webDefaultApl/topPage.cgi”」のように記述され、提供されるべき情報を表示するためのトップページが直接呼ばれるように設定される。

【0 1 0 6】

以下、画面がフレームによって分割されるような場合に端末 4 0 の端末種別に応じた情報提供が行われる処理について説明をする。

【0 1 0 7】

以後、ページ遷移が行われた際には、相対パス情報 6 0 b のみが指定されるため、共通パス情報 6 0 a に含まれる端末種別「PC」が常に継承されることになる。つまり、同一の共通パス情報 6 0 a を複数のWebアプリケーション 3 0 1 が参照する。

【0 1 0 8】

図 1 5 及び図 1 6 は、mainFrame.cgi による処理フローを示す図である。図 1 5 において、端末 4 0 は、メインフレーム HTML を要求する（メインフレーム HTML 要求）（ステップ S 3 1）。「/pc/admin/ja/webDefaultApl/mainFrame.cgi」が指定され、form データが添付された GET メソッドでメインフレーム HTML 要求が行われる。

【0 1 0 9】

情報処理装置 1 0 0 の HTTP デモン 2 は、HTTP 接続情報 3 4 0 とハッシュリストとをシーケンス制御ライブラリ 1 1 0 に通知する（ステップ S 3 2）。

【0 1 1 0】

この場合、HTTP デモン 2 によって、HTTP 接続情報 3 4 0 と関連付けられる HTTP 要求情報 3 5 0 のリクエストメソッドには「GET」が設定され、リクエストの URL 3 5 4 には「/pc/admin/ja/webDefaultApl/mainFrame.cgi」が指定される。この指定によって、以後、端末 4 0 から相対パスによって要求を受信した場合、「/pc/admin/ja」を示す共通パス情報 6 0 a が参照可能となる。従って、端末 4 0 の端末種別が「PC」であることが分かる。また、HTTP デモン 2 によって、ハッシュリストに受信した form データが設定される。

【0 1 1 1】

シーケンス制御ライブラリ 1 1 0 は、Web アプリ振分共通ライブラリ 1 2 0 を呼び出す（ステップ S 3 3）。その際、シーケンス制御ライブラリ 1 1 0 は、端末 4 0 との接続に関する情報を示す HTTP 接続情報と、ハッシュリストとを引数として設定する。

【0 1 1 2】

Web アプリ振分共通ライブラリ 1 2 0 は、URL に基づいて、「webDefaultApl」を示す Web アプリケーション ID 6 4 に対応する Web ページハンドラ 2 0 0 として、ハッシュリスト 5 0 a からデフォルトハンドラ 2 1 0（「default_handler」）を特定して呼び出す（ステップ S 3 4）。

【0 1 1 3】

Web ページハンドラ 2 0 0 では、デフォルトハンドラ 2 1 0 が、URL から

「mainFrame.cgi」を示すページ情報 6 5 を取得し、その「mainFrame.cgi」に対応する Web ページ機能 3 0 0 として、ハッシュリスト 5 0 b から mainFrame() 関数を特定して呼び出す (ステップ S 3 5)。この際、HTTP 接続情報が引数として設定される。例えば、Web ページ機能 3 0 0 が C 言語で開発されている場合、Web ページハンドラ 2 0 0 は、C 言語の処理型に変換して mainFrame() 関数をコールする。

【 0 1 1 4 】

mainFrame()関数の実行によって、初期画面 (topPage.cgi) を構成するフレーム情報が作成される (ステップ S 3 6)。そのフレーム情報が処理結果データとして、Web ページハンドラ 2 0 0 におけるデフォルトハンドラ 2 1 0 へ通知される (ステップ S 3 7)。

【 0 1 1 5 】

以下、ステップ S 3 8、S 3 9、S 4 0、S 4 1 及び S 4 2 による XML で記述するシリアライズの処理は、上記ステップ S 1 8、S 1 9、S 2 0、S 2 1 及び S 2 2 での処理と同様であるのでその説明を省略する。

【 0 1 1 6 】

Web アプリ振分共通ライブラリ 1 2 0 は、XML で記述される表示データを端末 4 0 の種別に応じた HTML の記述に変換するために、ハッシュリスト 5 0 b を参照して、「mainFrame.cgi」を示すページ情報 5 6 に対応するスタイルシート「mainFrame.xsl」を取得して、XSLT プロセッサ 2 0 5 に対して XSL 変換要求を行う (ステップ S 4 3)。この際、表示データ「Response DOM」と、スタイルシート「mainFrame.xsl」とが引数として設定される。

【 0 1 1 7 】

XSL 変換要求を受信した XSLT プロセッサ 2 0 5 は、引数で指定されたスタイルシート「mainFrame.xsl」を実行することによって、端末種別に応じたテンプレートをスタイルシートの記述から選択し、XML で記述された表示データ「Response DOM」の構文を解析し (XML パース)、DOM を作成し、要素及び属性を追加して (ステップ S 4 4)、XML ライブラリによって HTML の記述に変換する (ステップ S 4 5)。端末種別情報 6 3 a が「PC」を示す場合、画

面をフレームによって複数の画面に分割するHTMLが出力される。一方、端末種別情報63aが「PDA」を示す場合、直接トップページを呼び出すためのJava（登録商標）Script（登録商標）が記述されたHTMLが出力される。

【0118】

XSLTプロセッサ205は、変換したHTMLの記述をXSL変換結果としてWebアプリ振分共通ライブラリ120へ通知する（ステップS46）。

【0119】

ページ情報65が「mainFrame.cgi」の場合のWebページ機能300によってHTMLで記述された処理結果が、Webアプリ振分共通ライブラリ120にメインフレームHTMLとして出力される。メインフレームHTMLでは、メニューページを構成するフレームからは「menuPage.cgi」が呼び出され、トップページを構成するフレームからは「topPage.cgi」が呼び出され、ヘッダページを構成するフレームからは「headerPage.cgi」が呼び出されるように、HTMLで記述される。

【0120】

このメインフレームHTML応答は、順に、Webアプリ振分共通ライブラリ120からシーケンス制御ライブラリ110へ通知され（ステップS47）、シーケンス制御ライブラリ110からHTTPデーモン2へ通知される（ステップS48）。そして、HTTPデーモン2によって、HTTPに従ってメインフレームHTML応答がステップS31における端末40の要求に対する応答として送信され（ステップS49）、mainFrame.cgiによって実行される処理フローを終了する。

【0121】

端末40がクライアントPC41である場合のmainFrame.cgiによるメインフレームHTML出力について図17で説明する。図17は、PC用のメインフレームHTML出力の例を示す図である。図17において、メインフレームHTML410は、記述411によって端末40のWebブラウザ上に3つのフレームを作成し、各<frame>タグからページをCGIで呼び出す。

【 0 1 2 2 】

<frame noresize src="/webDefaultApl/headerPage.cgi" name="header" marginheight="0" marginwidth="0" scrolling="no">によって示される記述 4 1 2 は、フレーム内にヘッダページを表示する。「/webDefaultApl/headerPage.cgi」で示される記述 4 1 3 によって、

「webDefaultApl」に基づいてデフォルトハンドラ 2 1 0 が実行され、「headerPage.cgi」に基づいてW e b デフォルトアプリ 3 1 0 のheaderPage()関数が実行され、ヘッダページがフレーム内に表示される。

【 0 1 2 3 】

また、<frame src="/webDefaultApl/menuPage.cgi" name="menu"marginheight="0" marginwidth="0" scrolling="auto">によって示される記述 4 1 4 は、フレーム内にメニューページを表示する。「/webDefaultApl/menuPage.cgi」で示される記述 4 1 5 によって、「webDefaultApl」に基づいてデフォルトハンドラ 2 1 0 が実行され、「menuPage.cgi」に基づいてW e b デフォルトアプリ 3 1 0 のmenuPage()関数が実行され、メニューページがフレーム内に表示される。

【 0 1 2 4 】

更に、<frame src="/webDefaultApl/topPage.cgi" name="top" marginheight="0" marginwidth="0" scrolling="auto">によって示される記述 4 1 6 は、フレーム内にトップページを表示する。「/webDefaultApl/topPage.cgi」で示される記述 4 1 7 によって、「webDefaultApl」に基づいてデフォルトハンドラ 2 1 0 が実行され、「topPage.cgi」に基づいてW e b デフォルトアプリ 3 1 0 のtopPage()関数が実行され、トップページがフレーム内に表示される。

【 0 1 2 5 】

記述 4 1 2、4 1 4 及び 4 1 6 は、相対パス情報 6 0 b を指定するのみであるから、共通パス情報 6 0 a が U R L で指定されることになる。

【 0 1 2 6 】

図 1 7 において、端末 4 0 が携帯電話器 4 2 又は P D A 端末 4 3 である場合、フレームを構成せずに、t o p P a g e . c g i のみを呼び出すように構成される。例えば、H T M L の記述は、図 1 4 に示すデフォルト H T M L 出力と同様の

記述であるが、jumpToMainFrame()の代わりに、self.location.pathname="/pda/admin/ja/webDefaultApl/topPage.cgi" によって直接 topPage.cgi を呼び出すようなjumpToTopPage()が使用される。

【0127】

図18及び図19は、topPage.cgiによる処理フローを示す図である。図18において、端末40は、情報処理装置100に備えられたプロッタ35-1又はスキャナ35-2のステータスを表示するHTMLを要求する（トップページHTML要求）（ステップS51）。「topPage.cgi」が指定され、formデータが添付されたGETメソッドでトップページHTML要求が行われる。

【0128】

情報処理装置100のHTTPデーモン2は、HTTP接続情報340とハッシュリストとをシーケンス制御ライブラリ110に通知する（ステップS52）。この場合、HTTPデーモン2によって、HTTP接続情報340と関連付けられるHTTP要求情報350のリクエストメソッドには「GET」が設定され、リクエストのURLには「topPage.cgi」が指定される。また、HTTPデーモン2によって、ハッシュリストに受信したformデータを設定する。

【0129】

シーケンス制御ライブラリ110は、Webアプリ振分共通ライブラリ120を呼び出す（ステップS53）。その際、シーケンス制御ライブラリ110は、端末40との接続に関する情報を示すHTTP接続情報340と、ハッシュリストとを引数として設定する。

【0130】

Webアプリ振分共通ライブラリ120は、URLに基づいて、「webDefaultApl」を示すWebアプリケーションID64に対応するWebページハンドラ200として、ハッシュリスト50aからデフォルトハンドラ210（「default_handler」）を特定して呼び出す（ステップS54）。

【0131】

Webページハンドラ200では、デフォルトハンドラ210が、URLから

「topPage.cgi」を示すページ情報 6 5 を取得し、その「topPage.cgi」に対応する Web ページ機能 3 0 0 として、ハッシュリスト 5 0 b から topPage() 関数を特定して呼び出す（ステップ S 5 5）。この際、HTTP 接続情報 3 4 0 が引数として設定される。例えば、Web ページ機能 3 0 0 が C 言語で開発されている場合、Web ページハンドラ 2 0 0 は、C 言語の処理型に変換して topPage() 関数をコールする。

【0 1 3 2】

topPage() 関数の実行によって、Web デフォルトアプリ 3 1 0 は、初期画面（topPage.cgi）に機器（例えば、プリンタとしてのプロッタ 3 5 - 1）のステータスを表示するために、API 1 7 を介してネットワーク制御部 1 0 1 に対して、デバイス名、コメント等の情報を要求する（ステップ S 5 6 - 2）。ネットワーク制御部 1 0 1 は、その要求に応じてデバイス名、コメント等の情報を通知する（ステップ S 5 6 - 4）。

【0 1 3 3】

Web ページ機能 3 0 0 の Web デフォルトアプリ 3 1 0 は、例えば、プリンタのステータスを、API 1 7 を介してプリンタ制御部 1 0 3 に対して要求する（プリンタステータス要求）（ステップ S 5 6 - 6）。プリンタ制御部 1 0 3 は、プロッタ 3 5 - 1 の状態をプリンタステータスとして Web デフォルトアプリ 3 1 0 へ通知する（ステップ S 5 6 - 8）。

【0 1 3 4】

そして、Web デフォルトアプリ 3 1 0 は、そのプリンタステータスと、HTTP 要求情報 3 5 0 のリクエストの URL 3 4 5 から端末種別情報 6 3 a と、プロファイル情報 6 3 b と、言語情報 6 3 c とを、処理結果データとして、Web ページハンドラ 2 0 0 におけるデフォルトハンドラ 2 1 0 へ通知する（ステップ S 5 7）。

【0 1 3 5】

以下、ステップ S 5 8、S 5 9、S 6 0、S 6 1 及び S 6 2 による XML で記述するシリアルライズの処理は、上記ステップ S 1 8、S 1 9、S 2 0、S 2 1 及び S 2 2 での処理と同様であるのでその説明を省略する。

【 0 1 3 6 】

W e b アプリ振分共通ライブラリ 1 2 0 は、XML で記述される表示データを端末 4 0 の種別に応じた H T M L の記述に変換するために、ハッシュリスト 5 0 b を参照して、「topPage.cgi」を示すページ情報 5 6 に対応するスタイルシート「topPage.xml」を取得して、X S L T プロセッサ 2 0 5 に対して X S L 変換要求を行う（ステップ S 6 3）。この際、表示データ「Response DOM」と、スタイルシート「topPage.xml」とが引数として設定される。

【 0 1 3 7 】

X S L 変換要求を受信した X S L T プロセッサ 2 0 5 は、引数で指定されたスタイルシート「topPage.xml」を実行することによって、端末種別に応じたテンプレートをスタイルシートの記述から選択し、XML で記述された表示データ「Response DOM」の構文を解析し（XML パース）、DOM を作成し、要素及び属性を追加して（ステップ S 6 4）、XML ライブラリによって H T M L の記述に変換する（ステップ S 6 5）。

【 0 1 3 8 】

X S L T プロセッサ 2 0 5 は、変換した H T M L の記述を X S L 変換結果として W e b アプリ振分共通ライブラリ 1 2 0 へ通知する（ステップ S 6 6）。

【 0 1 3 9 】

ページ情報 6 5 が「topPage.cgi」の場合の W e b ページ機能 3 0 0 によって H T M L で記述された処理結果が、W e b アプリ振分共通ライブラリ 1 2 0 にトップページ H T M L として出力される。トップページ H T M L では、プロッタ 3 5 - 1 の状態に関する情報が H T M L で記述される。

【 0 1 4 0 】

このトップページ H T M L 応答は、W e b アプリ振分共通ライブラリ 1 2 0 からシーケンス制御ライブラリ 1 1 0 へ通知され（ステップ S 6 7）、シーケンス制御ライブラリ 1 1 0 から H T T P デーモン 2 へ通知される（ステップ S 6 8）。そして、H T T P デーモン 2 によって、H T T P に従ってトップページ H T M L 応答がステップ S 5 1 における端末 4 0 の要求に対する応答として送信され（ステップ S 6 9）、topPage.cgiによって実行される処理フローを終了する。

【0 1 4 1】

端末 4 0 がクライアント P C 4 1 である場合、上記同様にして、各フレームから呼び出されたHeaderPage.cgi及びmenuPage.cgiによる処理が実行され、フレームによって分割された画面がクライアント P C 4 1 の W e b ブラウザに表示される。一方、端末 4 0 が携帯電話器 4 2 又は P D A 端末 4 3 である場合、フレームによる画面の分割は行われないので、一つの画面上にトップページのみが表示される。

【0 1 4 2】

XML で記述された処理結果について説明する。図 2 0 は、XML で記述された t o p P a g e . c g i による処理結果の例を示す図である。

【0 1 4 3】

図 2 0 に示される XML 4 2 0 において、<networkResponse>から</networkResponse>までで示される記述 4 2 1 によって W e b デフォルトアプリ 3 1 0 の top Page()関数による処理結果が示される。<terminal>pc</terminal>で示される記述 4 2 2 は、端末 4 0 がクライアント P C 4 1 であることを示し、<language>ja</language>で示される記述 4 2 3 は、端末 4 0 の W e b ブラウザの言語が日本語であることを示し、<profile>admin</profile>で示される記述 4 2 4 は、端末 4 0 の利用者が管理者であることを示している。記述 4 2 2 から 4 2 4 は、U R L の共通パス情報 6 0 a の端末種別情報 6 3 a、プロフィール情報 6 3 b、言語情報 6 3 c に基づく情報である。

【0 1 4 4】

<deviceName>Printer 1</deviceName>で示される記述 4 2 5 は、機器名が「Printer 1」であることを示す。<comment>XXXXXXXXXX</comment>で示される記述 4 2 6 は、「Printer 1」に関するコメントが「XXXXXXXXXX」であることを示す。<status>Ready</status>で示される記述 4 2 7 は、「Printer 1」の状態が「正常」であることを示す。

【0 1 4 5】

W e b ページハンドラ 2 0 0 が、W e b ページ機能 3 0 0 よる処理結果データを XML で記述してシリアルライズする機能を有するため、W e b ページ機能 3 0

0の各Webアプリケーション301は、それぞれに端末種別に対応する処理部を有する必要がない。

【0146】

このようにXMLで記述されシリアル化された処理結果データを端末種別に基づいて、XSLTプロセッサ205によってHTMLに変換するためのスタイルシートについて図21で説明する。図21は、処理結果をXMLからHTMLに変換するためのXSLの記述例を示す図である。

【0147】

図21に示されるXSL430において、記述431によって、図20のXML420の記述422で示される<terminal>タグで囲まれた要素に「pda」の文字列が含まれているか否か、すなわち、XML420はPDA端末用の処理結果であるか否かが判断される。「pda」の文字列が含まれている場合は記述432によってPDA端末用のテンプレートが適用され、「pda」の文字列が含まれていない場合は記述433によって一般PC用のテンプレートが適用される。記述434にはPDA端末用のテンプレートが記述され、記述435には一般PC用のテンプレートが記述される。なお、それぞれのテンプレートについては図中では詳述していない。

【0148】

例えば、PDA端末用のテンプレートを記述する記述434では、携帯電話器42又はPDA端末43の表示画面のサイズを考慮して、PC用では表示されるイメージを表示しない、PCより小さいフォントサイズを指定する、PCより少ない文字数を指定するなどの記述となる。

【0149】

XSLによってXMLをHTMLに変換する機能をWebアプリ振分共通ライブラリ120が有することによって、一元的に端末種別に応じたHTMLの作成を行うことができる。よって、Webページ機能300における各Webアプリケーション301は、それぞれに端末種別に対応する処理部を有する必要がない。

【0150】

このようにXMLからHTMLへ変換するXSLを各Webアプリケーション301に応じて予め用意しておくことによって、それぞれの処理結果を示すHTMLを端末40に提供し、端末40のWebブラウザによってその処理結果が表示される。

【0151】

端末40がクライアントPC41の場合、Webデフォルトアプリ310のtopPage()関数の処理結果を示すHTMLは、例えば、図22に示されるように記述される。図22は、topPage.cgiの処理結果を示すPC用のHTMLの記述例を示す図である。図22において、PC用のHTML440は、端末40がクライアントPC41の場合に提供され、記述441によってWebデフォルトアプリ310のtopPage()関数の処理結果がクライアントPC41のWebブラウザに表示される。

【0152】

システムステータスを示す記述442は、「システムステータス」をフォントサイズ4で表示することを示している。を示す記述443は、イメージを表示することを示している。<p>システム名：Printer 1</p>を示す記述445は、「システム名：Printer 1」を表示することを示し、<p>コメント：XXXXXXXXXX</p>を示す記述446は、「コメント：XXXXXXXXXX」を表示することを示し、<p>システムの状態：正常</p>を示す記述447は、「システムの状態：正常」を表示することを示している。

【0153】

このようなPC用のHTML440によって、クライアントPC41のWebブラウザには、図23に示すような画面が表示される。図23は、クライアントPCでのシステムステータスの表示例を示す図である。図23において、mainFrame.cgiによって画面450が3つのフレームで構成され、各フレームには、headerPage.cgiによってヘッダページが表示され、menuPage.cgiによってメニューページが表示され、topPage.cgiによってトップページが表示される。

【0 1 5 4】

トップページには、図 2 2 に示す HTML 2 4 0 の記述 4 4 2 によって「システムステータス」を示すテキスト 4 5 2 が表示され、記述 4 4 4 によってイメージ 4 5 4 が表示され、記述 4 4 5 によって「システム名：P r i n t e r 1」を示すテキスト 4 5 5 が表示され、記述 4 4 6 によって「コメント：X X X X X X X X X」を示すテキスト 4 5 6 が表示され、記述 4 4 7 によって「システムの状態：正常」を示すテキスト 4 5 7 が表示される。

【0 1 5 5】

また、ヘッダページを表示するフレームには、例えば、「A A A A A A」等のヘッダとしての情報が表示される。メニューページを表示するフレームには、例えば、選択可能なメニュー項目として「システムステータス」、「ネットワーク設定」、「機器構成情報」等のテキスト 4 5 9 が表示される。

【0 1 5 6】

「ネットワーク設定」の HTML の記述は、例えば、`<p>ネットワーク設定</p>` のように記述される。この URL には、相対パス情報 6 0 b のみが設定されているため、共通パス情報 6 0 a がそのまま継承され、「ネットワーク設定」に関する情報は PC 用にトップページに表示される。つまり、Web アプリケーション ID を示す「webNetsettingApl」から相対パスとして設定されており、「../」の記述によって現在の端末種別情報 5 3 a が継承される。また、「target=" top"」の記述によって、「ネットワーク設定」に関する情報は PC 用にトップページに表示される。

【0 1 5 7】

「機器構成情報」の HTML の記述についても同様に、`<p>機器構成情報</p>` のように記述される。従って、この URL には、相対パス情報 6 0 b のみが設定されているため、共通パス情報 6 0 a がそのまま継承され、「機器構成情報」に関する情報は PC 用にトップページに表示される。

【0 1 5 8】

クライアント P C 4 1 が本発明に係る情報処理装置 1 0 0 に対して情報を要求した場合、画面 4 5 0 のように表示されるが、携帯電話器 4 2 又は P D A 端末 4 3 が同じ情報を要求した場合、図 2 4 に示すような P D A 端末用の H T M L が作成され、図 2 5 のような画面が P D A 端末 4 3 に表示される。

【 0 1 5 9 】

図 2 4 は、 t o p P a g e . c g i の処理結果を示す P D A 端末用の H T M L の記述例を示す図である。図 2 4 において、P D A 端末用の H T M L 4 4 0 a は、端末 4 0 が P D A 端末 4 3 の場合に提供され、記述 4 4 1 a によって W e b デフォルトアプリ 3 1 0 の topPage() 関数の処理結果が P D A 端末 4 3 の W e b ブラウザに表示される。

【 0 1 6 0 】

システムステータスを示す記述 4 4 2 a は、「システムステータス」をフォントサイズ 3 で表示することを示している。記述 4 4 2 a では、P D A 端末 4 3 用に文字フォントが小さく設定される。

【 0 1 6 1 】

P D A 端末用の H T M L 4 4 0 a には、P C 用の H T M L 4 4 0 の記述 4 4 3 に相当する記述が存在しない。

【 0 1 6 2 】

P C 用の H T M L 4 4 0 の記述 4 4 5 及び 4 4 6 と同様に、<p>システム名：P r i n t e r 1</p>を示す記述 4 4 5 a は、「システム名：P r i n t e r 1」を表示することを示し、<p>コメント：X X X X X X X X X</p>を示す記述 4 4 6 a は、「コメント：X X X X X X X X X」を表示することを示す。

【 0 1 6 3 】

クライアント P C 4 1 に比べて表示領域の小さい P D A 端末 4 3 では、P C 用の H T M L 4 4 0 の記述 4 4 7 に相当する記述 4 4 7 a において、「システムの状態：正常」を表示する代わりに、例えば、「状態：正常」を表示することを示し、表示文字数を少なくして表示する。

【 0 1 6 4 】

また、P D A 端末 4 3 では画面をフレームによって複数に分割することが適切

でないため、クライアント P C 4 1 では他のフレームに表示される情報を、P D A 端末 4 3 では 1 つの画面（ページ）内に表示する必要がある。記述 4 4 8 a は、P C 用ではメニューページを表示する H T M L に記述されるものであって、P D A 端末用の H T M L 4 4 0 a では、`<p>ネットワーク設定</p>`及び`<p>機器構成情報</p>`のように記述し、クライアント P C 4 1 のメニューページに表示される選択可能な項目のうち「ネットワーク設定」と「機器構成情報」とを表示することを示す。記述 4 4 8 a において、P C 用の H T M L 4 4 0 では、表示すべきフレームを`target=" top"`によって指定していたが、その記述が削除される。

【 0 1 6 5 】

このような P D A 端末用の H T M L 4 4 0 a によって、P D A 端末 4 3 の W e b ブラウザには、図 2 5 に示すような画面が表示される。図 2 5 は、P D A でのシステムステータスの表示例を示す図である。図 2 5 において、画面 4 5 0 a は、フレームによって分割されない 1 つの画面で構成される。

【 0 1 6 6 】

画面 4 5 0 a には、図 2 4 に示す H T M L 2 4 0 a の記述 4 4 2 a によって「システムステータス」を示すテキスト 4 5 2 a が表示され、クライアント P C 4 1 には表示されるイメージを表示することなく、記述 4 4 5 a によって「システム名：P r i n t e r 1」を示すテキスト 4 5 5 が表示される。記述 4 4 6 a によって「コメント：X X X X X X X X X」を示すテキスト 4 5 6 a が表示され、記述 4 4 7 a によってクライアント P C 4 1 より文字数の少ない「状態：正常」を示すテキスト 4 5 7 a が表示される。

【 0 1 6 7 】

更に、クライアント P C 4 1 ではメニューページに表示される選択可能な「ネットワーク設定」及び「機器構成情報」が、記述 4 4 8 a によって選択可能な「ネットワーク設定」及び「機器構成情報」を示すテキスト 4 5 8 a として表示される。

【 0 1 6 8 】

画面 450 a が携帯端末 42 又は PDA 端末 43 に表示された状態で、PDA 端末 43 のユーザが「ネットワーク設定」を選択すると、共通パス情報 60 a が継承され相対パス情報 60 b のみが「ネットワーク設定」を示す URL に変更されるため、図 26 に示すような画面が携帯端末 42 又は PDA 端末 43 に表示される。図 26 は、PDA 端末でのネットワーク設定の表示例を示す図である。

【0169】

図 26 において、画面 460 a は、画面 450 a からページ遷移して表示された画面であって、ページ遷移時に共通パス情報 60 a が継承されるため、携帯端末 42 又は PDA 端末 43 の表示画面のサイズに応じて「ネットワーク設定」のページが表示される。携帯端末 42 又は PDA 端末 43 の利用者は、表示画面のサイズに合わせて「ネットワーク設定」のページを表示させるための設定を何ら行なわない。

【0170】

従来のような端末 40 の端末種別が識別されない情報処理装置に対して、ページ遷移を行うと、クライアント PC 41 への表示画面のサイズに応じた表示が成されるため、図 27 に示すような情報提供となってしまう。端末種別情報 53 a が継承されないため、大き過ぎる文字で表示されてしまい、携帯端末 42 又は PDA 端末 43 の表示画面に適した情報を提供することができない。

【0171】

本願発明によれば、HTTP デモン 2 が、端末 40 からの最初の要求を受信した時に default.cgi を呼び出すように URL を構成するため、default.cgi によって共通パス情報 60 a を持った URL を情報処理装置 100 にて自動的に設定することができる。従って、端末 40 の利用者は、表示画面を意識した情報の入力を行う必要がない。

【0172】

共通パス情報 60 a を相対パス情報 60 b の前に構成するため、ページからのリンクは相対パスを指定するのみで良く、HTML の作成を容易とすることができる。

【0173】

また、情報処理装置 100 は、スタイルシートを用いて、共通パス情報 60 a の端末種別情報 53 a に基づいて、XML から HTML への変換を行う。従って、情報処理装置 100 において、Web アプリケーション 301 の開発者は、端末種別を意識したアプリケーション開発をする必要がなく、開発者の負担を軽減することができる。つまり、各 Web アプリケーション 301 は、端末種別の違いに依存せずに、統一した処理結果を出力することができる。例えば、表示画面のサイズの違い、フレームによる表示画面の分割等が、Web アプリケーション 301 の開発に影響を与えることがない。

【0174】

上記実施例において、端末種別として、クライアント PC 41 を「PC」、携帯電話器 42 及び PDA 端末 43 を「PDA」として分類したが、表示画面に応じた分類が可能であれば良いため、「PC」、「PDA」の他に更に多くの端末種別があっても良い。また、分類は表示画面の特性に応じた分類であればよい。

【0175】

また、本願発明は、端末 40 として巨大スクリーンが情報処理装置 100 に接続される場合にも適応することができる。つまり、巨大スクリーンに応じて、文字を大きく変更したり、文字数を多くして情報量を多くしたりすることが可能となる。また、より大きいイメージ又は複数のイメージを表示して、効果的に情報を提供することができる。

【0176】

このような端末 40 に応じた情報を提供するために、本発明に係る情報処理装置 100 では、各スタイルシート 430 の端末種別に応じたテンプレートにて、Web ページ機能 300 から提供される処理結果の表示形式を調整及びイメージ等の付加を行なっている。

【0177】

【発明の効果】

以上、説明してきたように、本願発明によれば、端末からの要求時に端末種別が共通パス情報として URL の一部に設定されるため、複数の Web アプリケーション間でページ遷移した場合においても、端末の Web ブラウザに常に同一の

端末種別に対応した情報提供を行うことができる。また、処理結果のXMLによる記述と、端末種別を判断してXMLからHTMLへ変換する処理とを、複数のWebアプリケーションから切り離して構成することができる。

【図面の簡単な説明】

【図 1】

本発明の一実施例に係る情報処理装置のハードウェア構成を示すブロック図である。

【図 2】

情報処理装置の機能構成を示すブロック図である。

【図 3】

URL 構成を示す図である。

【図 4】

Web アプリケーションの実行を説明するためのブロック図である。

【図 5】

図 5 は、ハッシュリストを示す図である。

【図 6】

default.cgi による処理フローを示す図である。

【図 7】

default.cgi による処理フローを示す図である。

【図 8】

端末情報の解析処理について説明するためのフローチャート図である。

【図 9】

HTTP 接続情報のデータ構造を示す図である。

【図 1 0】

HTTP 要求情報のデータ構造を示す図である。

【図 1 1】

ヘッダ情報リストを示す図である。

【図 1 2】

Web ブラウザ種別—端末種別対応表を示す図である。

【図 1 3】

Web ブラウザ OS - 端末種別対応表を示す図である。

【図 1 4】

デフォルト HTML 出力の例を示す図である。

【図 1 5】

mainFrame.cgi による処理フローを示す図である。

【図 1 6】

mainFrame.cgi による処理フローを示す図である。

【図 1 7】

PC 用のメインフレーム HTML 出力の例を示す図である。

【図 1 8】

topPage.cgi による処理フローを示す図である。

【図 1 9】

topPage.cgi による処理フローを示す図である。

【図 2 0】

XML で記述された topPage.cgi による処理結果の例を示す図である。

【図 2 1】

処理結果を XML から HTML に変換するための XSL の記述例を示す図である。

【図 2 2】

topPage.cgi の処理結果を示す PC 用の HTML の記述例を示す図である。

【図 2 3】

クライアント PC でのシステムステータスの表示例を示す図である。

【図 2 4】

topPage.cgi の処理結果を示す PDA 端末用の HTML の記述例を示す図である。

【図 2 5】

P D A 端末でのシステムステータスの表示例を示す図である。

【図 2 6】

P D A 端末でのネットワーク設定の表示例を示す図である。

【図 2 7】

一般 P C 用のページを P D A 端末にそのまま表示した例を示す図である。

【符号の説明】

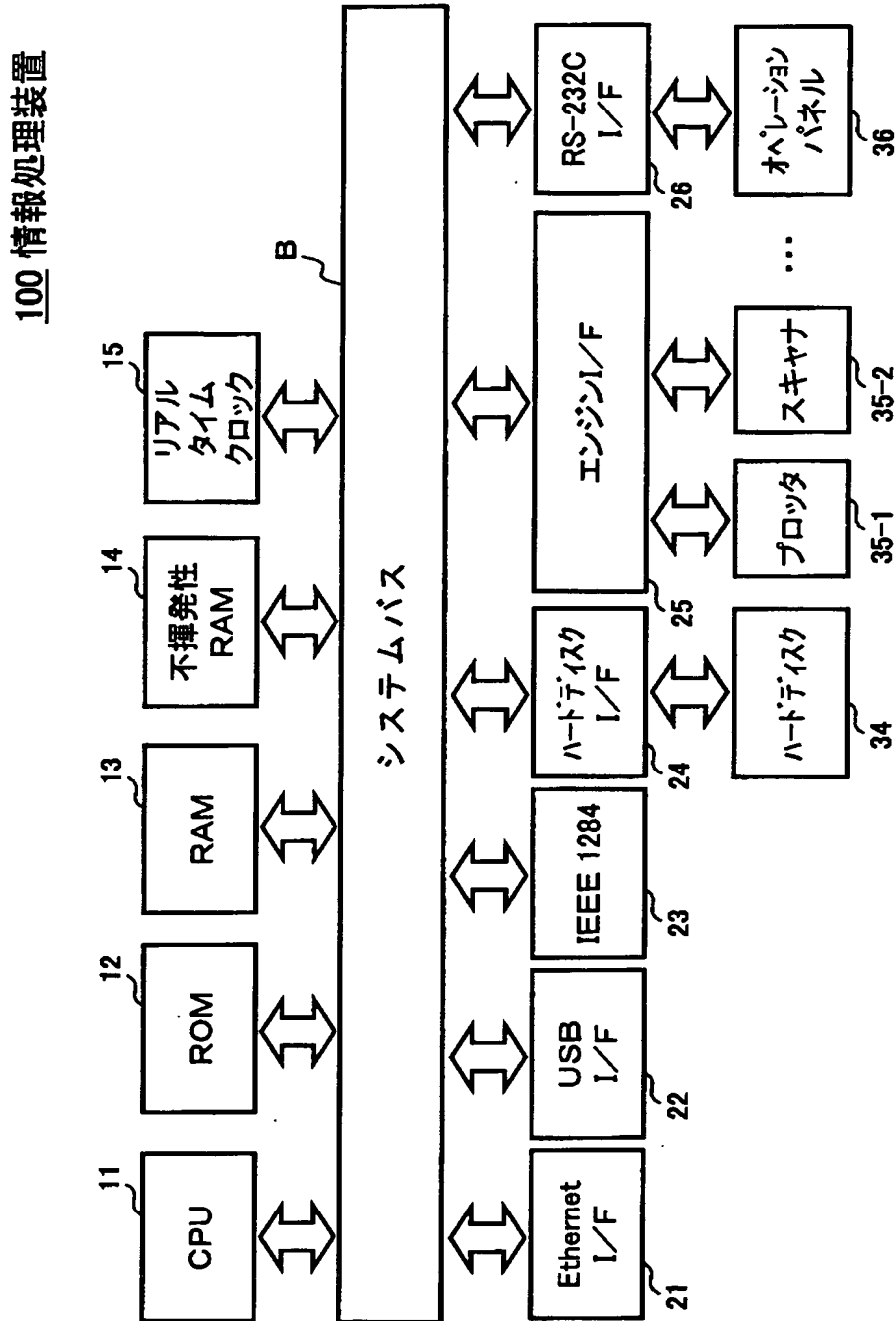
2	H T T P デーモン
1 6	インターネット
1 7	A P I
4 0	端末
4 1	クライアント P C
4 2	携帯電話器
4 3	P D A
1 0 0	情報処理装置
1 0 1	ネットワーク制御部
1 0 3	プリンタ制御部
1 0 5	スキャナ制御部
1 1 0	シーケンス制御ライブラリ
1 2 0	W e b アプリ振分共通ライブラリ
2 0 0	W e b ページハンドラ
2 0 1	S O A P ライブラリ
2 0 3	X M L ライブラリ
2 0 5	X S L T プロセッサ
3 0 0	W e b ページ機能 (W P F)

【書類名】

図面

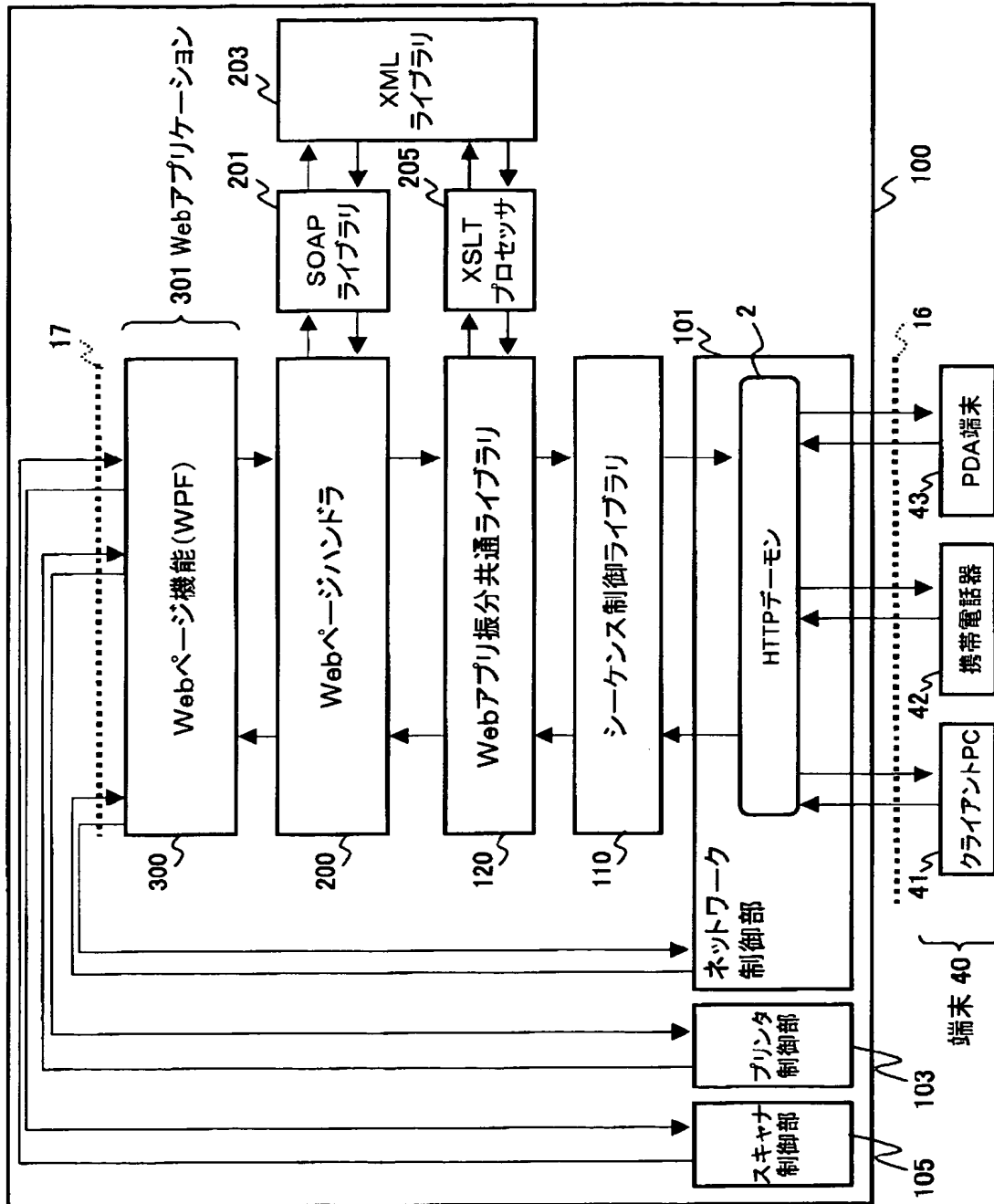
【図 1】

本発明の一実施例に係る情報処理装置の
ハードウェア構成を示すブロック図



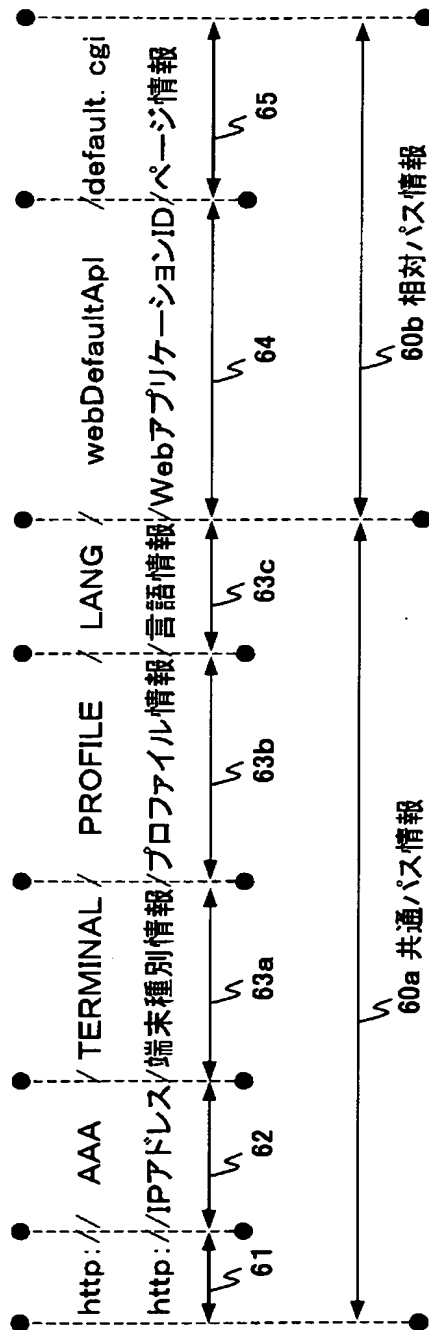
【図 2】

情報処理装置の機能構成を示すブロック図



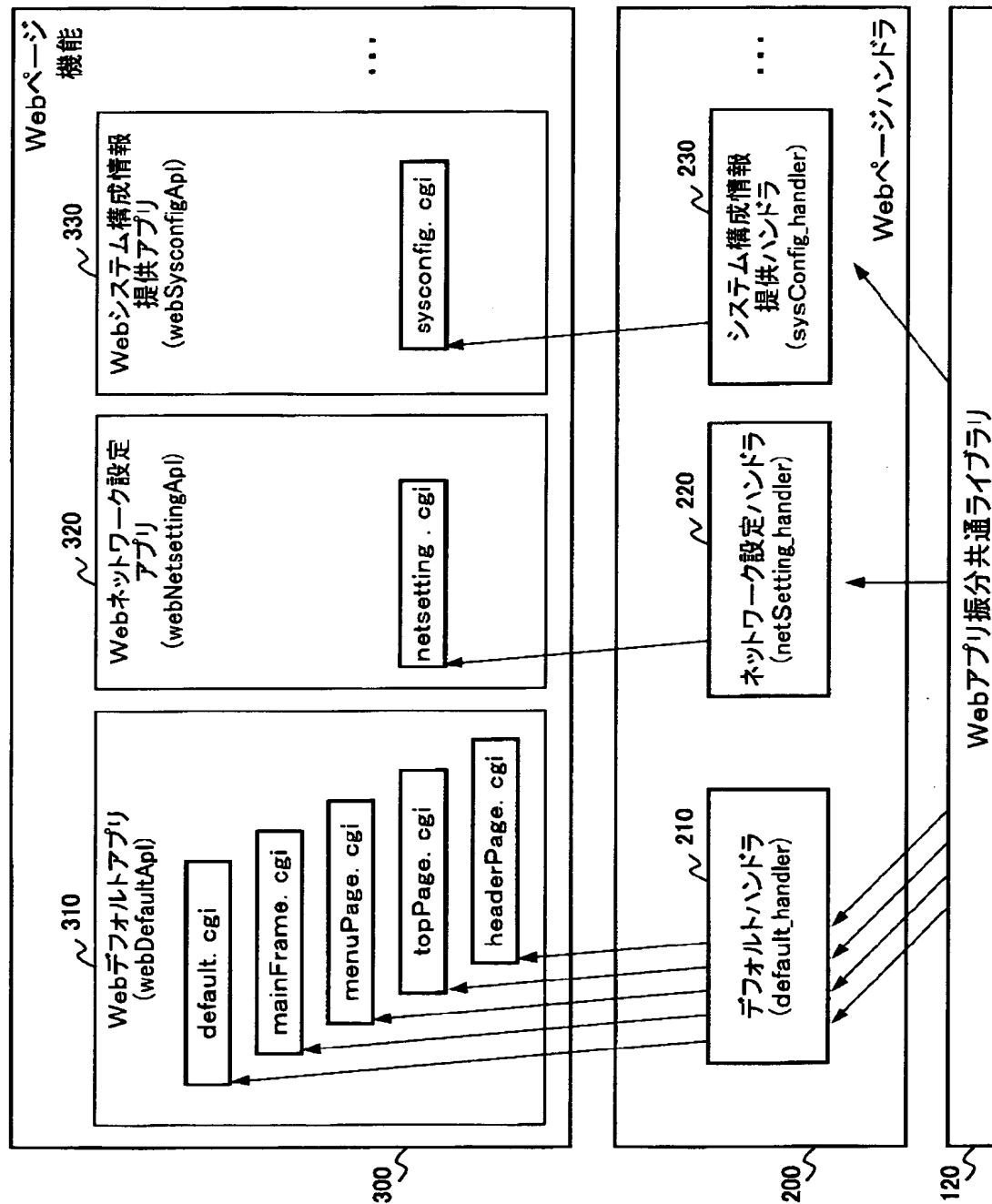
【図 3】

URL構成を示す図



【図 4】

Webアプリケーションの実行を説明するためのブロック図



【図 5】

ハッシュリストを示す図

(A)

50a

WebアプリケーションID	Webアプリケーションを呼ぶための Webページハンドラ
webDefaultApl	default__handler
webNetsettingApl	netSetting__handler
webSysconfigApl	sysConfig__handler
...	...

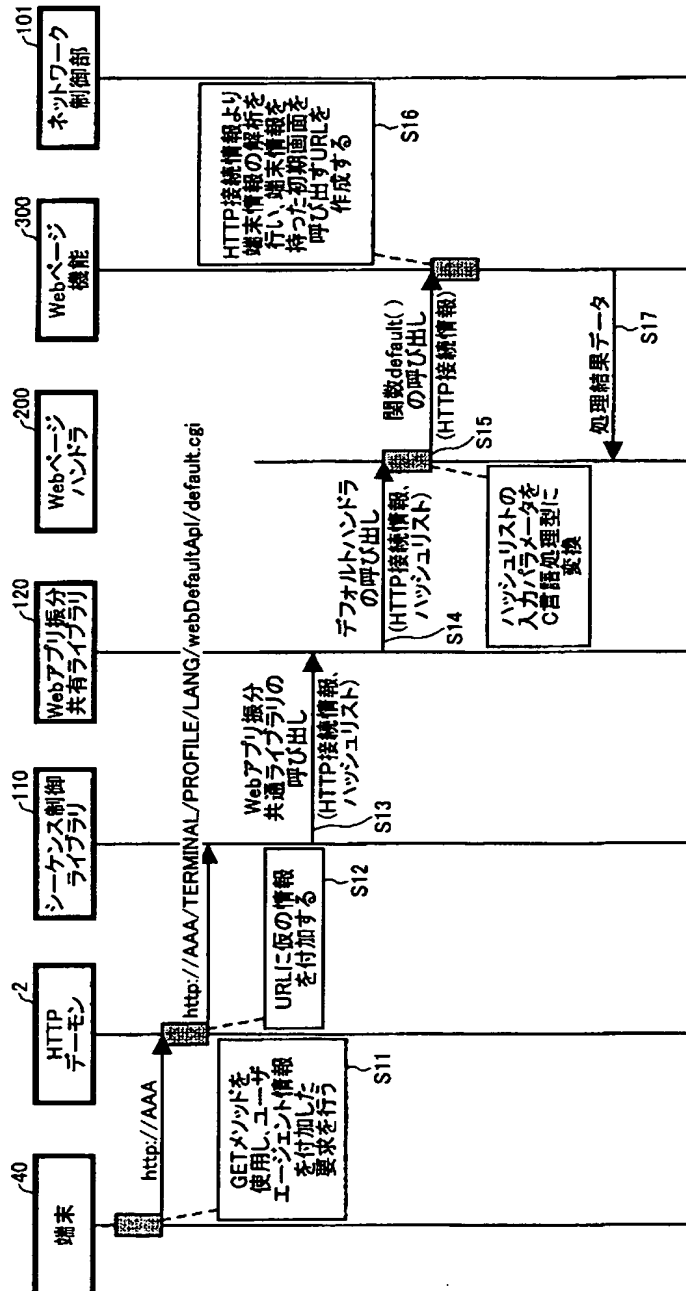
(B)

50b

Webページ機能ID	関数	スタイルシート
default. cgi	default()	default. xsl
mainFrame. cgi	mainFrame()	mainFrame. xsl
menuPage. cgi	menuPage()	menuPage. xsl
topPage. cgi	topPage()	topPage. xsl
headerPage. cgi	headerPage()	headerPage. xsl
netsetting. cgi	netsetting()	netsetting. xsl
sysconfig. cgi	sysconfig()	sysconfig. xsl
...

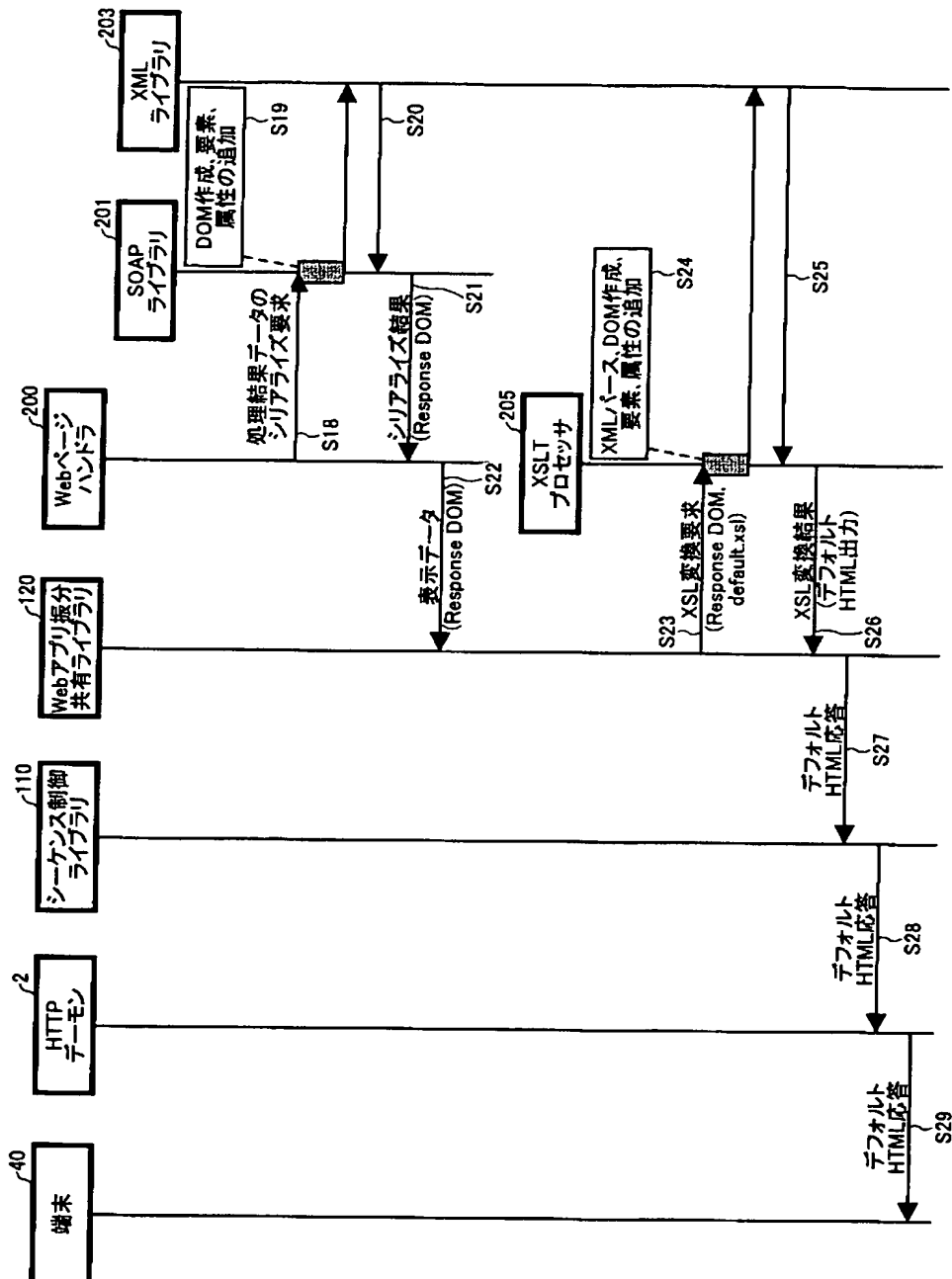
【図 6】

default. cgiによる処理フローを示す図

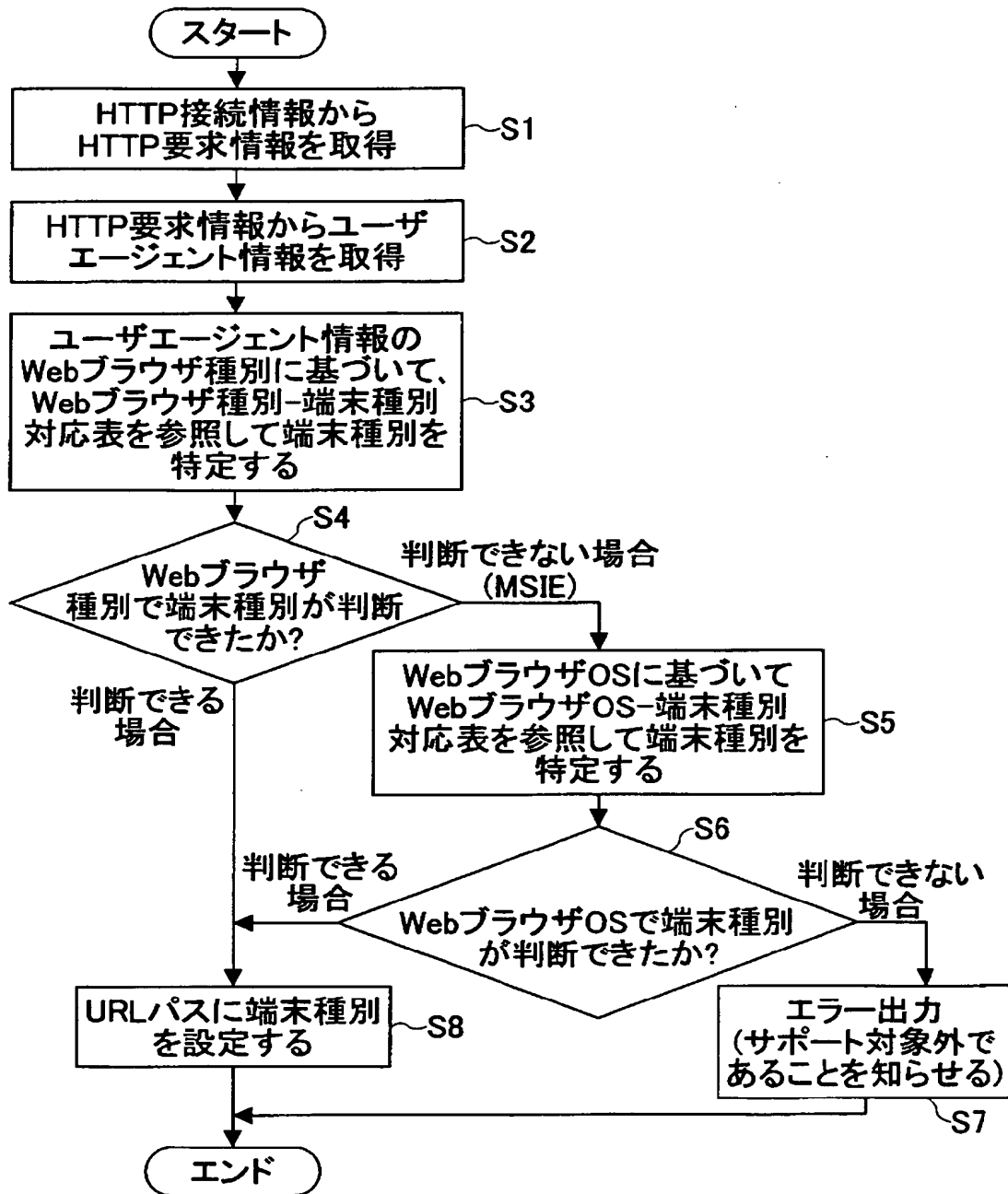


【図 7】

default. cgiによる処理フローを示す図



【図 8】

端末情報の解析処理について
説明するためのフローチャート図

【図 9】

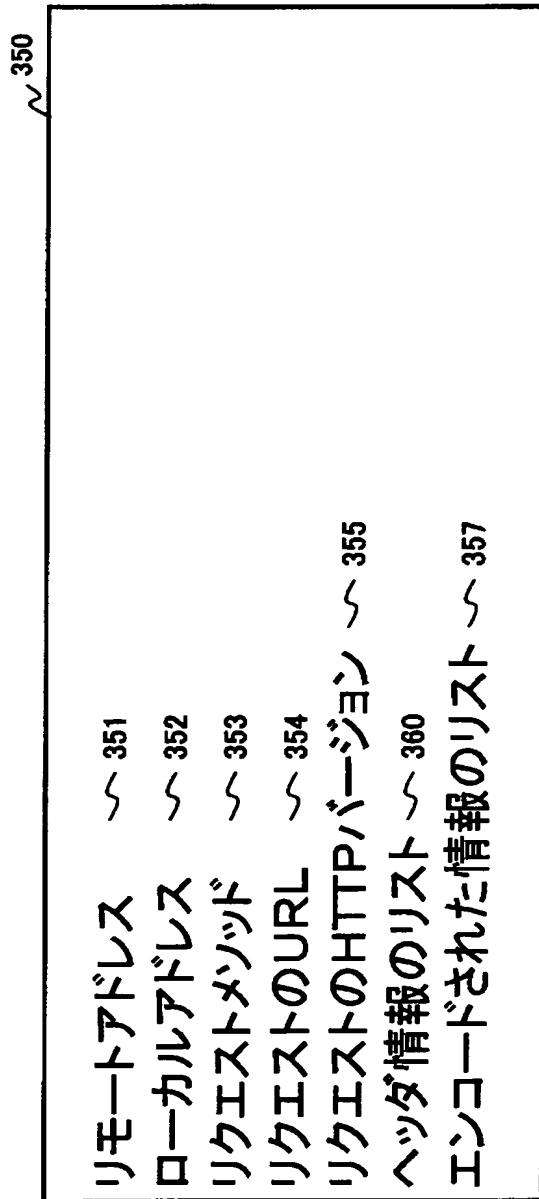
HTTP接続情報のデータ構造を示す図

~ 340

```
struct_HttpConnection {  
    HttpConnectionManager    *cm;  
    int    receiveId;  
    int    state;  
    buf_id_t    bufId; /* データを読み込むバッファ */  
    HTTP_REQUEST_INFO    *request; ~ 346  
    HTTP_RESPONSE_INFO    *response; ~ 347  
    pthread_mutex_t    mutexForEvent;  
    pthread_cond_t    condVarForEvent;  
};
```


【図 10】

HTTP要求情報のデータ構造を示す図



【図 11】

ヘッダ情報リストを示す図

GET HTTP/1.0
Accept:image/gif,image/xbitmap,image/jpeg,image/png,application/vnd.spowerpoint,application/
Voln.ms-excel,application/msword,application/pdf,*/
accept-Language:en-us;q=0.9;ja;q=0.7;it;q=0.6;sp;q=0.4;pt;q=0.3
User-Agent:Mozilla/4.0[compatible;MSIE 6.0;Windows NT 5.0;Q312461;NET CLR 1.0.3705 ~ 365
Host:133.139.49.79
Cookie:cookieOnOffCheck=on
Connection:keep-alive
Accept-encoding:gzip,deflate

【図 1 2】

Web ブラウザ種別—端末種別対応表を示す図

370

Webブラウザ種別	端末種別
DOCOMO	IMODE
xiino	PDA
MSPIE	PDA
HandHTTP	PDA
MSIE	O
Mozilla	PC
mozilla/5	PC
Netscape6	PC

【図 13】

WebブラウザOS—端末種別応表を示す図

380

WebブラウザOS	端末種別
Windows3.1	PC
Windows95	PC
Windows98	PC
WindowsME	PC
WindowsNT	PC
Windows2000	PC
WindowsXP	PC
Mac 68k	PC
Mac PowerPC	PC
Solaris	PC
Linux	PC
FreeBSD	PC
AIX	PC
IRIX	PC
HP	PC
OS/2	PC
WindowsCE	PDA

【図 14】

デフォルトHTML出力の例を示す図

```
<html>
<head>
<title>default page</title>
<script language="javascript">
<!--
function jumpToMainFrame() {
self.document.cookie="cookieOnOffChecker=on; path=/";
self.location.pathname="/pc/admin/ja/webDefaultAp/mainFrame.cgi";
}
// -->
</script>
</head>
<body onLoad="jumpToMainFrame()">
</body>
</html>
```

400 ~

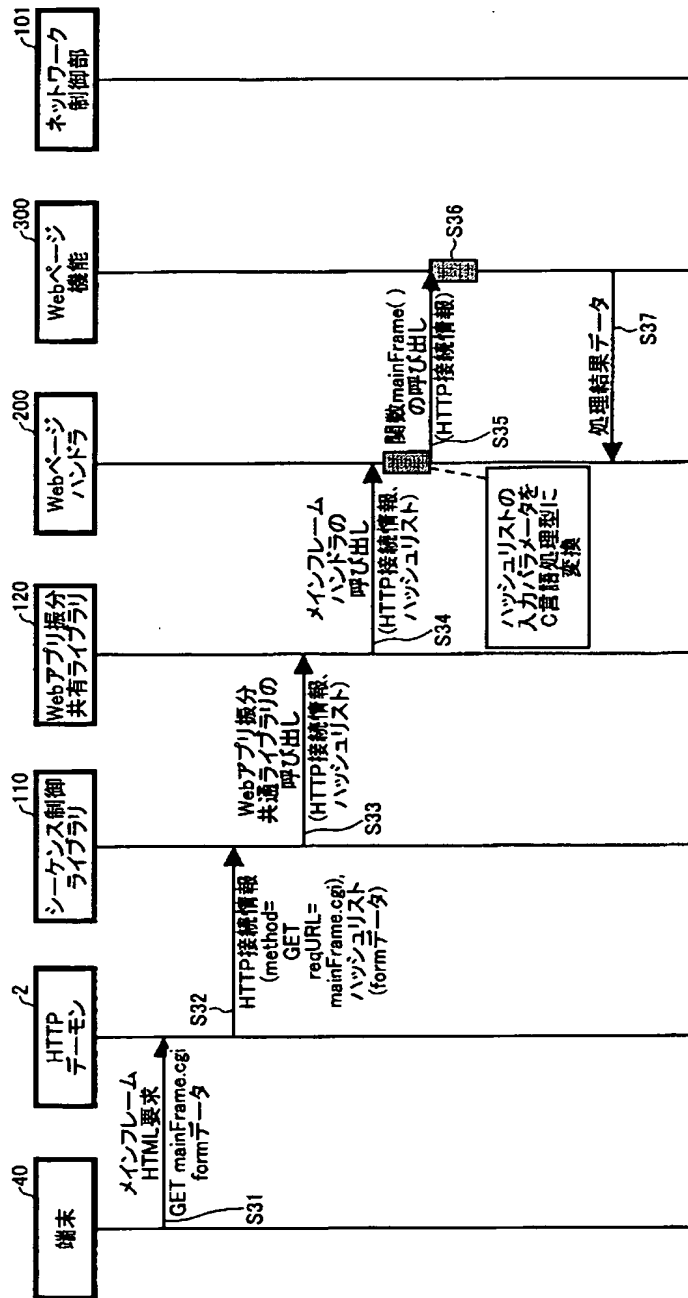
401 {

402

403 ~

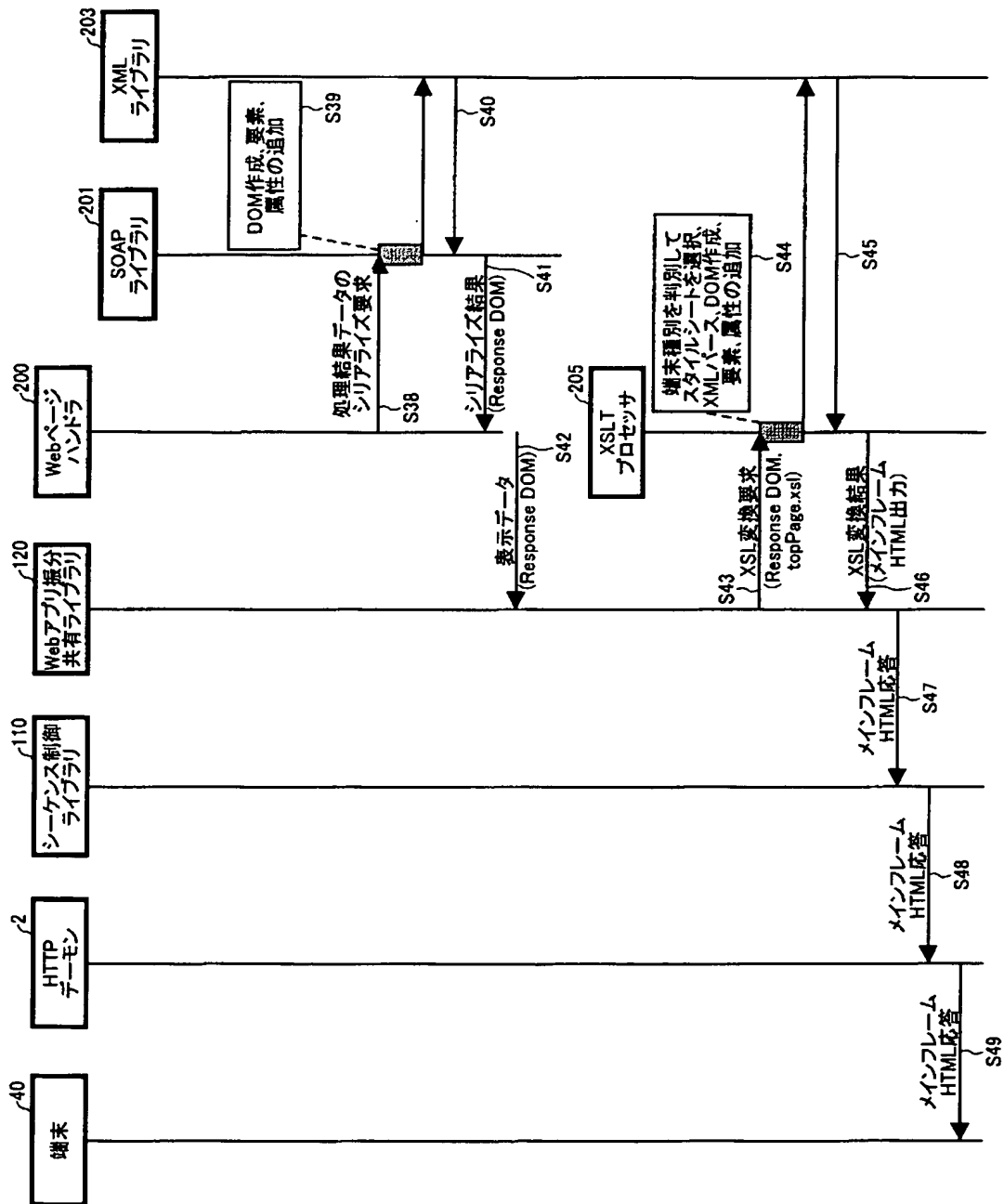
【図 15】

mainFrame.cgiによる処理フローを示す図



【図 16】

mainFrame.cgiによる処理フローを示す図



【図 17】

PC用のメインフレームHTML出力の例を示す図

```

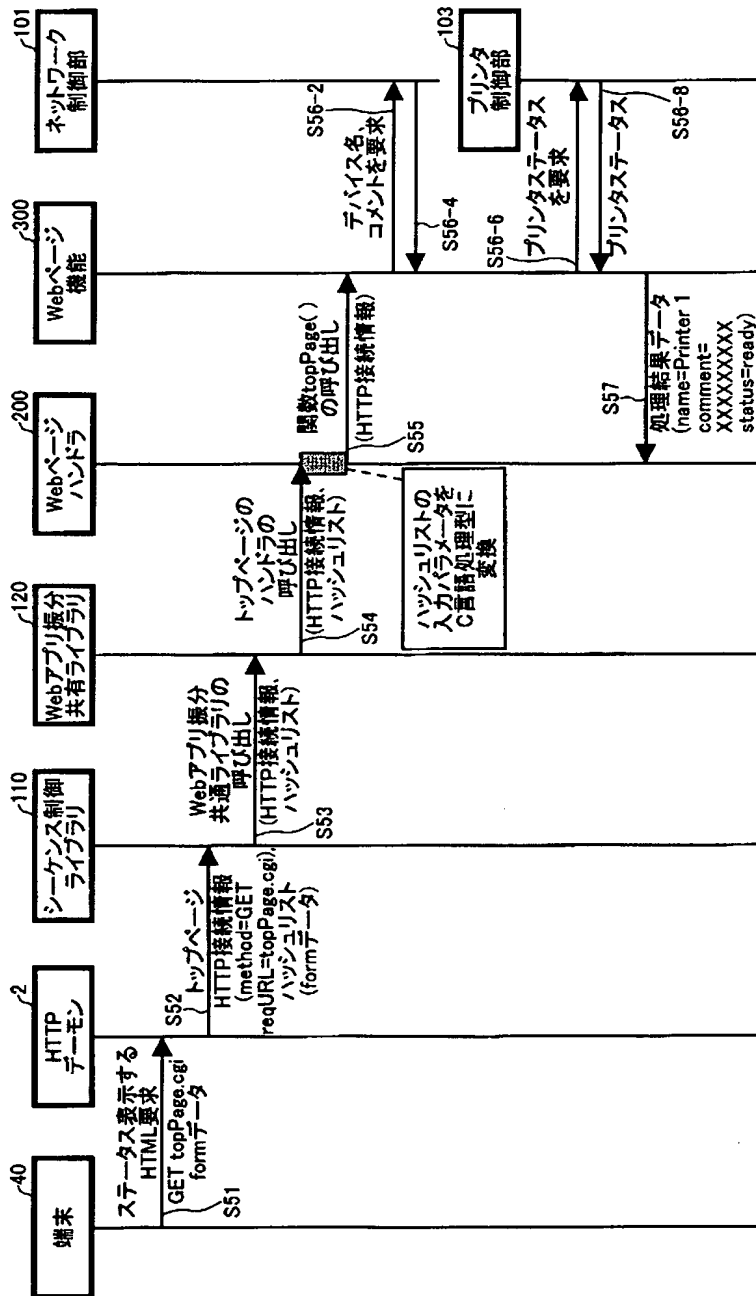
~ 410
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<html>
<head>
<meta HTTP-EQUIV="content-type" CONTENT="text/html; charset=UTF-8">
<title>Web Image Monitor</title>
</head>
<frameset rows="40,*" frameborder="0" border="0" framespacing="0">
  <frame noresize src="/webDefaultApl/headerPage.cgi" name="header" ~ 412
    marginheight="0" marginwidth="0" scrolling="no">
  <frameset cols="160,*" frameborder="1" border="1" framespacing="0">
    <frame src="/webDefaultApl/menuPage.cgi" name="menu" ~ 415
      marginheight="0" marginwidth="0" scrolling="auto"> ~ 414
    <frame src="/webDefaultApl/topPage.cgi" name="top" ~ 417
      marginheight="0" marginwidth="0" scrolling="auto"> ~ 416
    </frameset>
  </frameset>
</html>

```

Diagram illustrating the PC main frame HTML output example. The code is enclosed in a box with line numbers 410 to 417. Brackets on the right side group the code into sections: 411 (frameset rows), 412 (frame noresize), 413 (frame noresize src), 414 (frame src), 415 (frame src), 416 (frame src), and 417 (frame src).

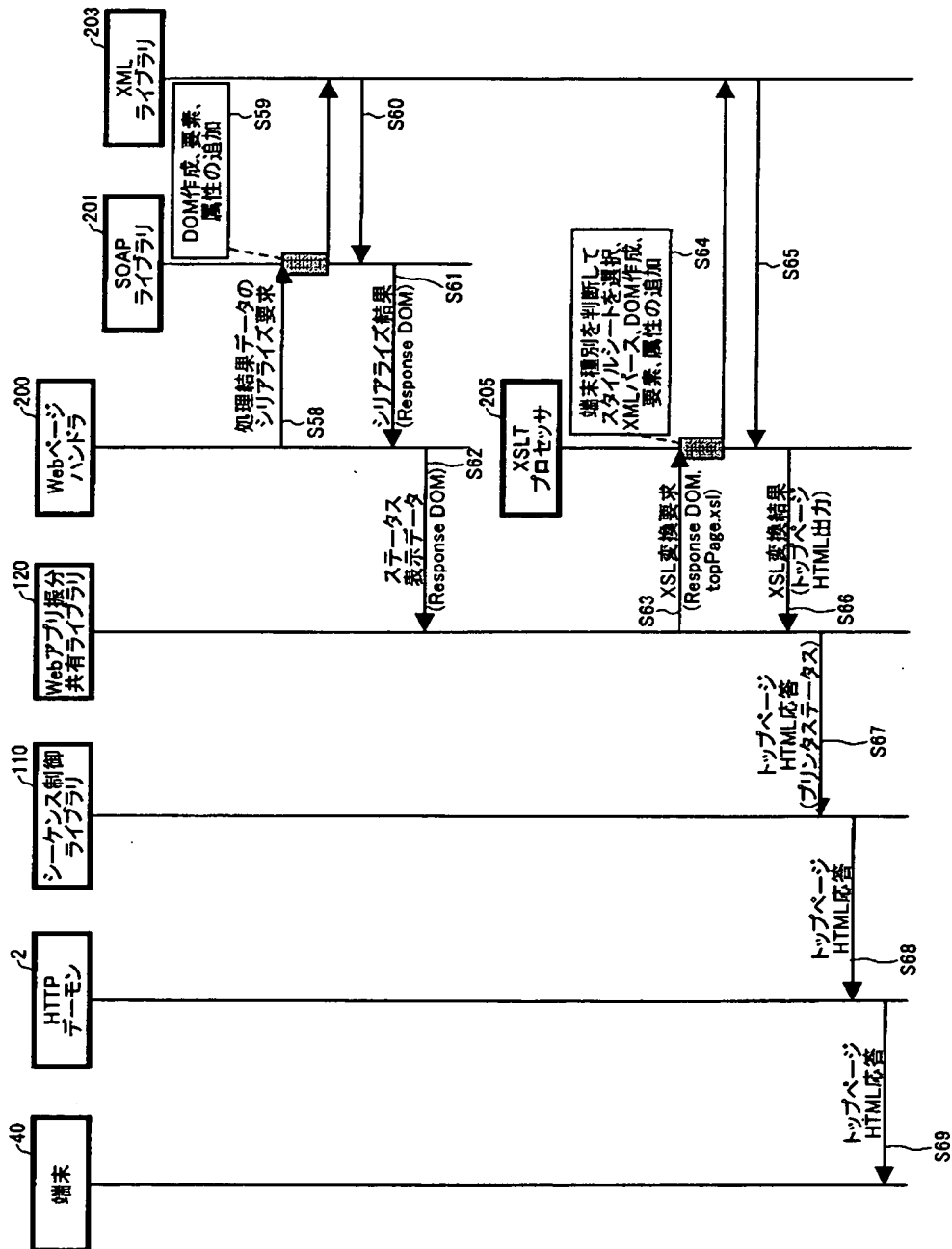
【図 18】

topPage.cgiによる処理フローを示す図



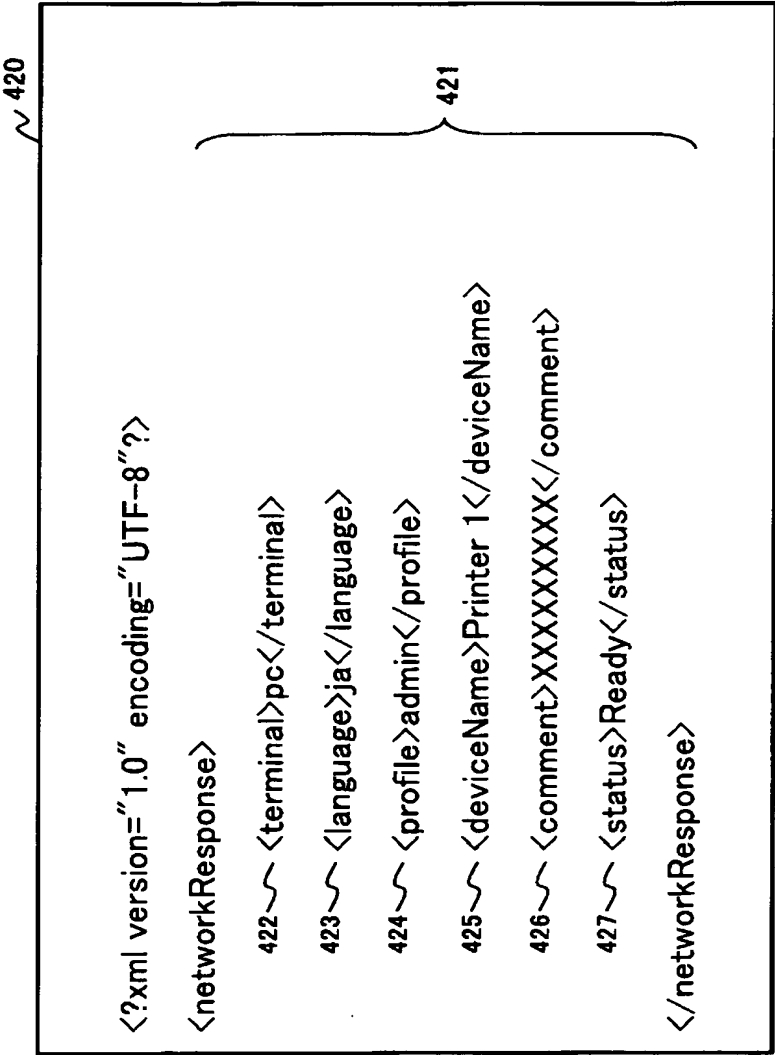
【図 19】

topPage.cgiによる処理フローを示す図



【図 2 0】

XMLで記述されたtopPage.cgiによる
処理結果の例を示す図



【図 21】

処理結果をXMLからHTMLに変換するための
XSLの記述例を示す図

430

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:sf="http://www.rr
rr.co.jp/xmlns/xslt/rdh/common" version="1.0">
  <xsl:output method="html" encoding="UTF-8" />
  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates />
      </body>
    </html>
  </xsl:template>

  <!-- Templates for body -->
  <xsl:template match="networkResponse">
    <!-- Display the page title -->
    <xsl:call-template name="statusTitle4CommonProfile" />
    <!-- Check current terminal type(pda or pc) and call suitable template -->
    <xsl:choose>
      <xsl:when test="contains(/urldevice, 'pda')"> ~ 431
        <xsl:call-template name="networkResponse_pda" /> ~ 432
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name="networkResponse" /> ~ 433
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>

  <!-- 共通項目 -->
  <xsl:template name="networkTitle4CommonProfile">
    <!-- 略 -->
  </xsl:template>

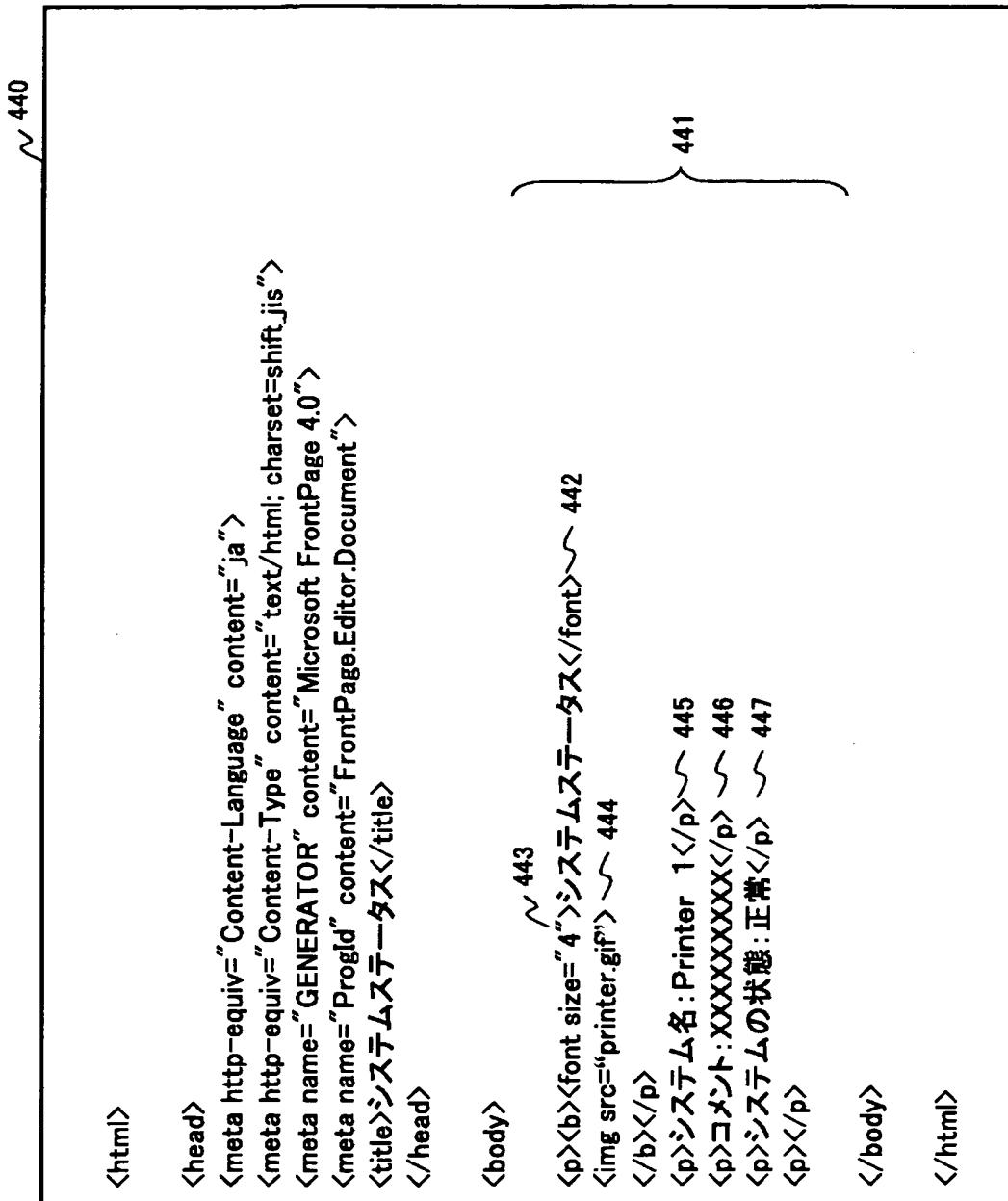
  <!-- PDA用 スタイルシート -->
  <xsl:template name="networkResponse_pda">
    <!-- 略 -->
  </xsl:template> } 434

  <!-- PC用 スタイルシート -->
  <xsl:template name="networkResponse">
    <!-- 略 -->
  </xsl:template> } 435
</xsl:stylesheet>

```

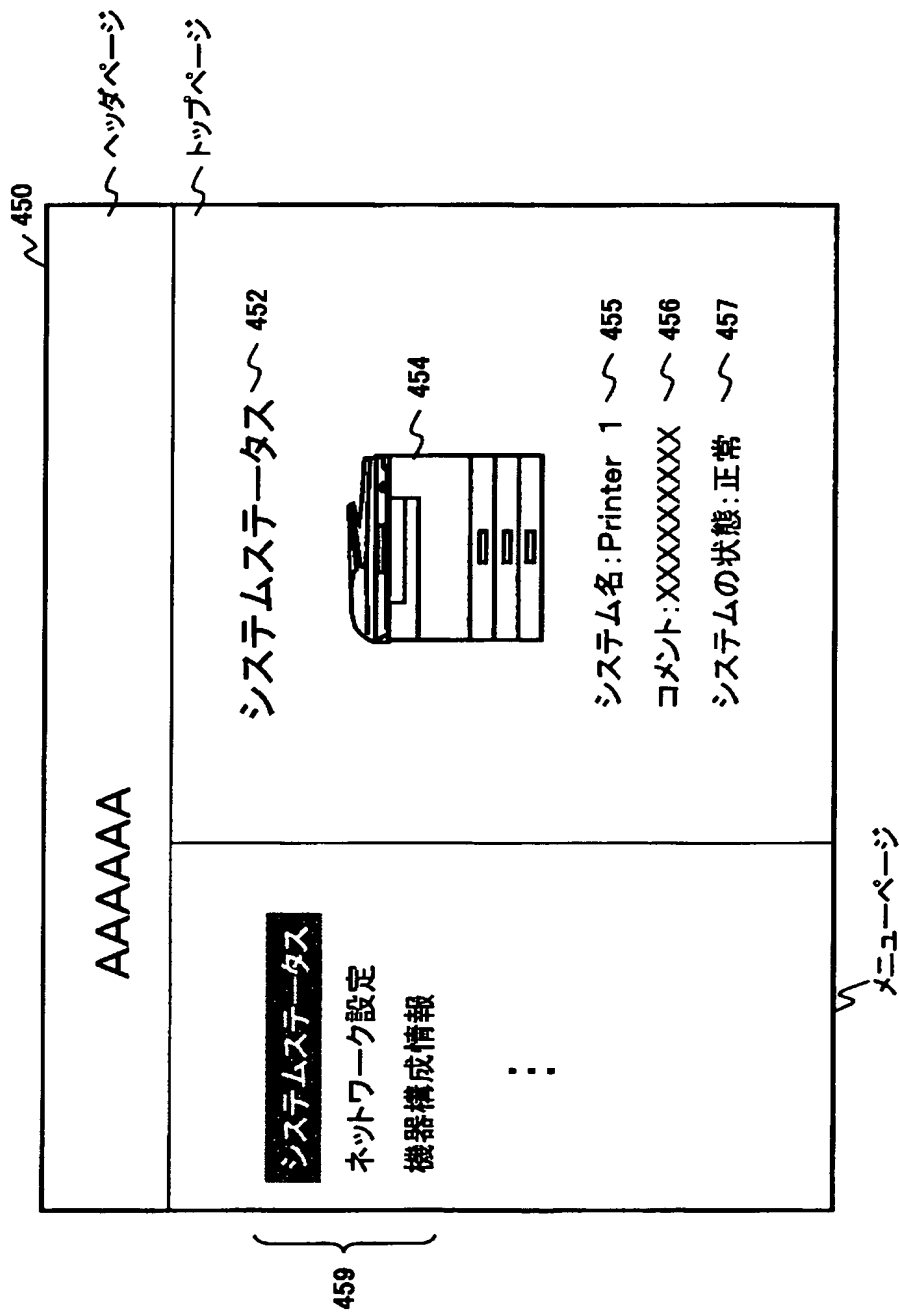
【図 22】

topPage.cgiの処理結果を示す
PC用のHTMLの記述例を示す図



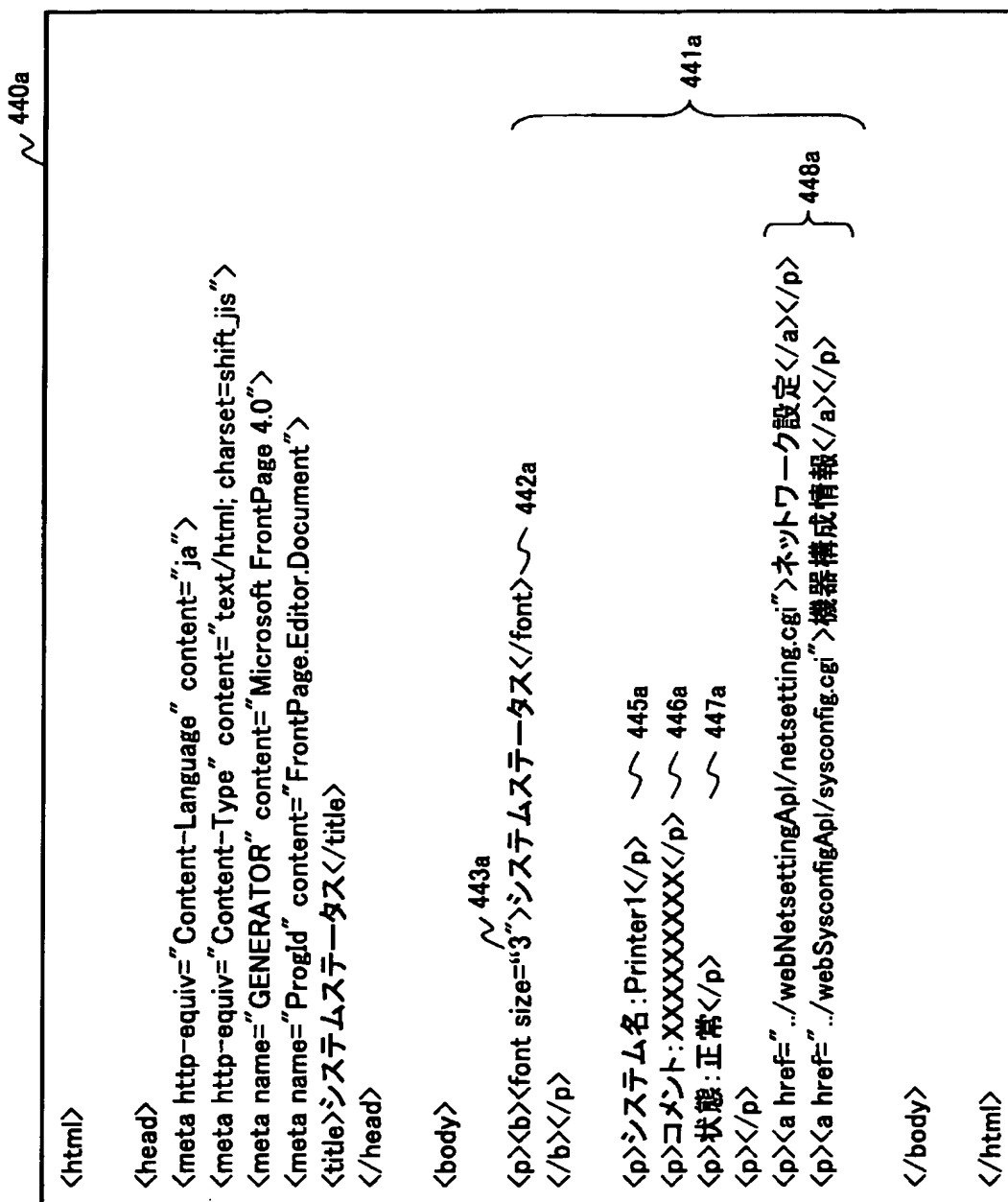
【図 2 3】

クライアントPCでのシステムステータスの表示例を示す図



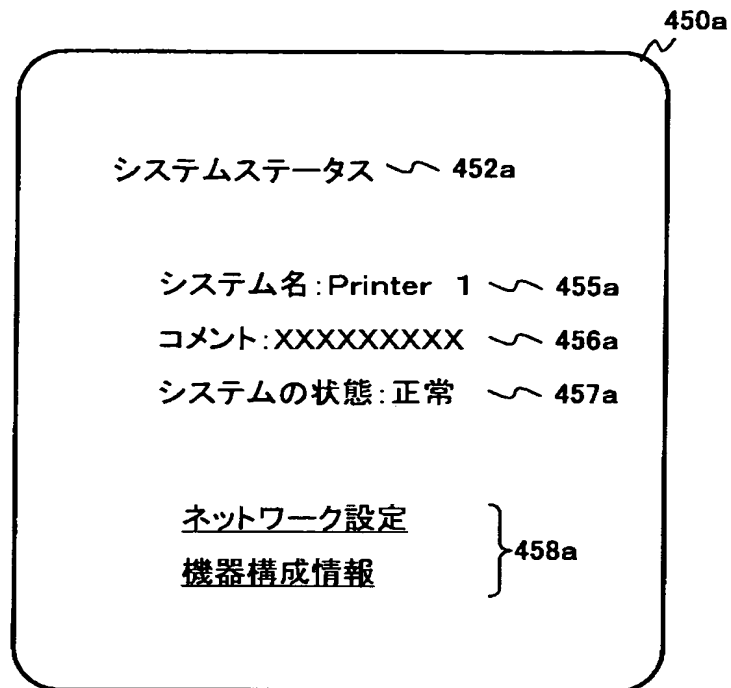
【図 24】

topPage.cgiの処理結果を示す
PDA端末用のHTMLの記述例を示す図



【図 25】

PDA端末でのシステムステータスの表示例を示す図



【図 2 6】

PDA端末でのネットワーク設定の表示例を示す図

460a

ネットワーク設定

ホスト名: PrinterXX

IPアドレス: 999.999.999.999

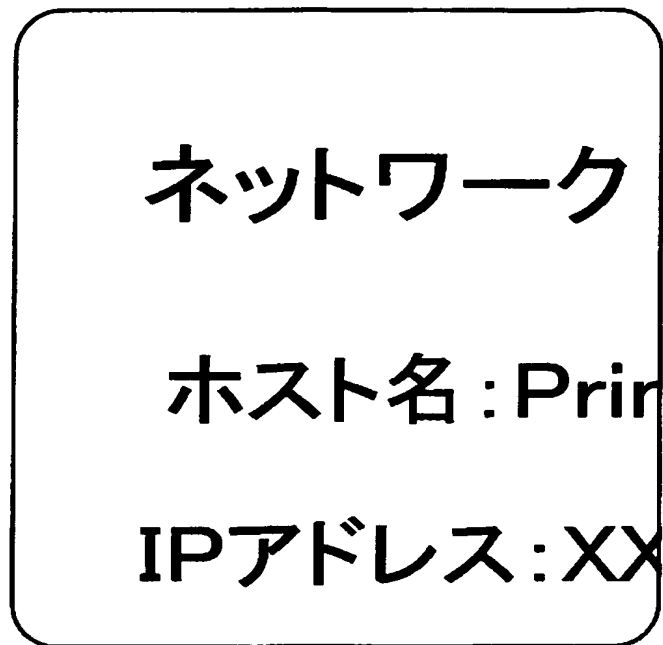
サブネットマスク: 255.255.255.0

システムステータス

機器構成情報

【図 2 7】

一般PC用のページをPDAにそのまま表示した例を示す図



【書類名】 要約書

【要約】

【課題】 本発明の課題は、W e b アプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数のW e b アプリケーションを有する情報処理装置を提供することを目的とする。

【解決手段】 本発明の課題は、ネットワークを介して接続される端末から第一要求を受信すると、該端末の端末種別を特定し、その特定した端末種別を示す端末種別情報を付加した該第一要求に応じたW e b 情報へのパスを基準パスとして該端末側からアクセスさせる基準W e b 情報を生成する基準W e b 情報生成手段と、上記基準W e b 情報を上記端末の上記第一要求に対する応答として送信することによって、上記基準パスによって該端末から上記W e b 情報を要求する第二要求を受信するようにした情報処理装置によって達成される。

【選択図】 図 6

特願 2 0 0 3 - 1 9 7 8 5 0

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 6 7 4 7]

1. 変更年月日 1 9 9 0 年 8 月 2 4 日
 [変更理由] 新規登録
 住 所 東京都大田区中馬込 1 丁目 3 番 6 号
 氏 名 株式会社リコー

2. 変更年月日 2 0 0 2 年 5 月 1 7 日
 [変更理由] 住所変更
 住 所 東京都大田区中馬込 1 丁目 3 番 6 号
 氏 名 株式会社リコー